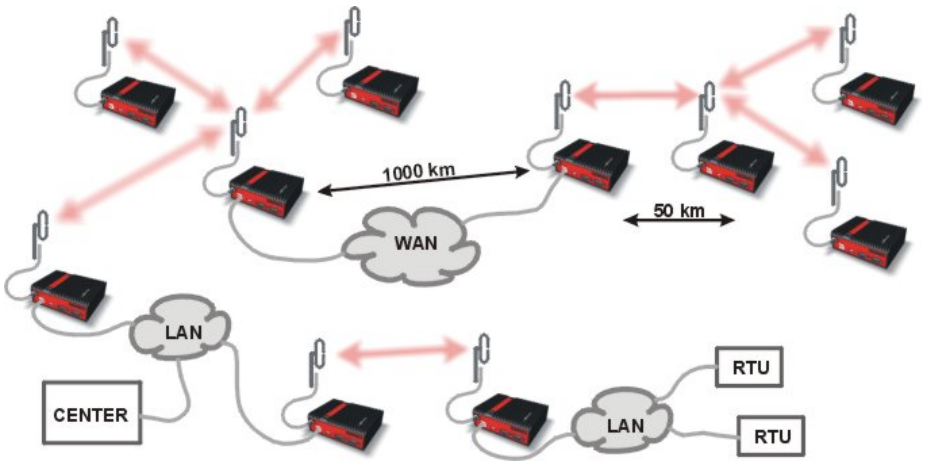
RACOM
RADIO DATA NETWORKS

Application notes



RipEX

version 1.5
4/11/2014
fw 1.3.x.x

RACOM s.r.o. • Mirova 1283 • 592 31 Nove Mesto na Morave • Czech Republic
Tel.: +420 565 659 511 • Fax: +420 565 659 512 • E-mail: racom@racom.eu

www.racom.eu

Table of Contents

1. Address planning 5
    1.1. End devices connected via serial interface 5
    1.2. End devices connected over Ethernet 8
    1.3. Ethernet addressing 9
2. SNMP 12
    2.1. Simple Network Management Protocol 12
    2.2. SNMP in RipEX 13
    2.3. Network Management System – ZABBIX 16
    2.4. How do I Know that Something Has Happened to the RipEX Station? 28
    2.5. What Else does Zabbix Offer? 35
    2.6. RipEX MIB Table 38
3. Data speed and Modulations 49
    3.1. Narrowband radio transmitter 49
    3.2. Narrowband radio receiver 53
    3.3. Conclusion 56
4. Autospeed 58
5. Back-to-Back repeater 60
    5.1. Back to Back in Bridge mode 60
    5.2. Back to Back in Router mode 60
6. Combining MORSE and RipEX networks 62
    6.1. RipEX part in Bridge mode 62
    6.2. RipEX in Router mode 63
7. Profibus 65
    7.1. Bridge and Router modes 65
    7.2. Profibus settings 66
    7.3. RipEX settings 68
    7.4. Advanced settings 69
8. Modbus TCP/RTU 71
    8.1. Modbus RTU 71
    8.2. Modbus TCP 72
    8.3. Modbus TCP, local TCP/IP connection 73
    8.4. Master - Modbus TCP, slaves - Modbus RTU 74
    8.5. Master Modbus TCP, slaves Modbus RTU or Modbus TCP 75
    8.6. Multiple Modbus TCP or Modbus RTU Masters and Slaves 75
9. UNI protocol 77
    9.1. MASTER – SLAVE communication 77
    9.2. MASTER – SLAVE with several Masters 79
    9.3. MASTER – MASTER 79
    9.4. MASTER UNI – ASYNC LINK SLAVES 79
10. Channel access 81
    10.1. Collisions 81
    10.2. Bridge mode 82
    10.3. Bridge mode and COM stream 85
    10.4. Router Mode 86
11. ARP Proxy & VLAN 88
    11.1. Introduction 88
    11.2. Transparent LAN (ARP Proxy) 88
    11.3. Transparent VLAN 89
    11.4. Configuration Examples 90
    11.5. Summary 103
12. Backup routes 104

12.1. Introduction 104
12.2. Backup Routing Management Protocol 104
12.3. Configuration Examples 105
12.4. Summary 117
A. Revision History 118

1. Address planning

In Router mode standard IP routing is used between individual RipEX radio modems and their interfaces. The only non-standard feature is that even if you assign all RipEX's radio interface IP addresses to a single network, the RipEX's may not "hear" each other over the radio channel; therefore, routing tables should include even routes within the same network (over repeaters).

This Application Note draws attention to certain situations in which routing tables can be simplified significantly.

1.1. End devices connected via serial interface

Every RipEX radio modem has two network interfaces, and hence two IP addresses. First is the Ethernet interface, second the radio interface. Serial interfaces are defined by their UDP port and are shared for the entire RipEX modem; as a result both RipEX IP addresses can be used to access them (both IP addresses work equally well).

The destination IP address of a packet received via the serial interface is determined inside the radio modem from the "SCADA address" depending on the protocol used, either using a mask or table (see RipEX manual, Adv. config., Protocols[1]). The source IP is generated similarly.

If all devices are connected to RipEX's via serial interface, it is helpful to only use the radio IP addresses for translation and routing of data. Ethernet IP addresses may be assigned randomly (you could keep their defaults, however we recommend setting Ethernet addresses similar to radio IP addresses to keep things organised). Remote service access over the radio channel is also possible via the IP addresses of the radio interface.

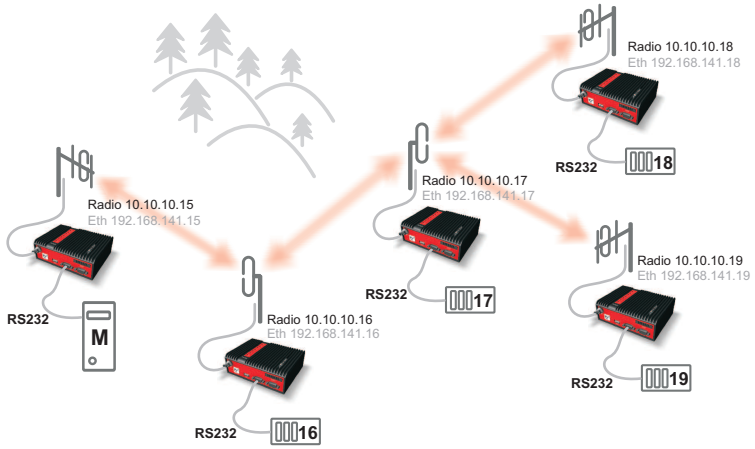


Fig. 1.1: Network 1

The following paragraph shows routing tables for individual radio modems which enable mutual communication between all devices. All destinations share the mask 255.255.255.255, i.e. 10.10.10.xx/32, interface Auto or Radio:

- For 10.10.10.15

```
Destination via Gateway
10.10.10.17 via 10.10.10.16
```

[1] http://www.racom.eu/eng/products/m/ripex/h-menu.html#protocols

```
10.10.10.18 via 10.10.10.16
10.10.10.19 via 10.10.10.16
```

- For 10.10.10.16

```
10.10.10.18 via 10.10.10.17
10.10.10.19 via 10.10.10.17
```

- For 10.10.10.17

```
10.10.10.15 via 10.10.10.16
```

- For 10.10.10.18

```
10.10.10.15 via 10.10.10.17
10.10.10.16 via 10.10.10.17
10.10.10.19 via 10.10.10.17 (this record is only necessary if you require
                             communication between end devices 19 and 18)
```

- For 10.10.10.19

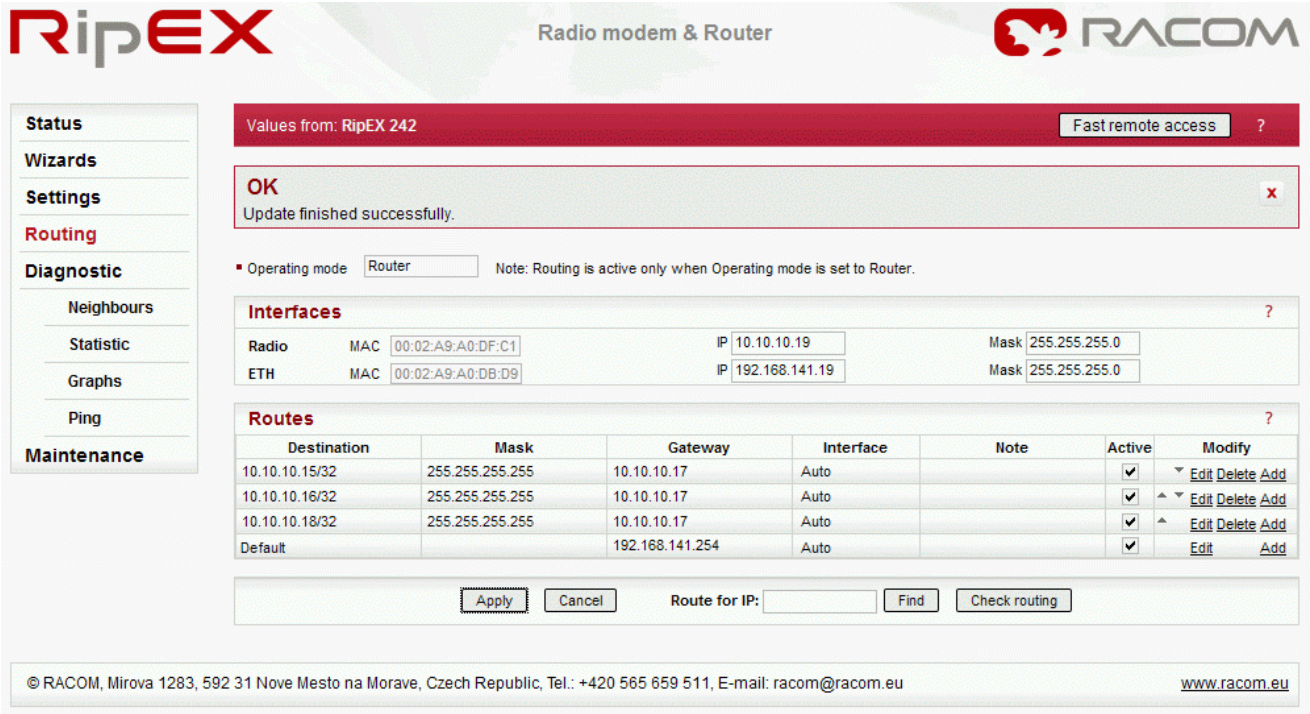
```
10.10.10.15 via 10.10.10.17
10.10.10.16 via 10.10.10.17
10.10.10.18 via 10.10.10.17 (this record is only necessary if you require
                             communication between end devices 19 and 18)
```

To display the full routing table type "ip route show table normal" in CLI interface

- For 10.10.10.19

```
10.10.10.15 via 10.10.10.17 dev radio  proto static
broadcast 10.10.10.0 dev radio  proto static  scope link  src 10.10.10.19
broadcast 10.10.10.255 dev radio  proto static  scope link  src 10.10.10.19
10.10.10.16 via 10.10.10.17 dev radio  proto static
10.10.10.18 via 10.10.10.17 dev radio  proto static
10.10.10.0/24 dev radio  proto static  scope link
192.168.141.0/24 dev eth0  proto static  scope link
default via 192.168.141.254 dev eth0  proto static
```

An example of a routing table on page Routing for 10.10.10.19



If SCADA device addresses can be chosen arbitrarily, routing can be significantly simplified when radio IP addresses can be grouped to subnets according to radio network layout.

One example of simplification is shown with repeaters connecting to separate subnets. The routing table can then contain a single record for all devices on the subnet.

In this example the first repeater connects to subnet 10.10.10.0/29, i.e. devices may have addresses from 10.10.10.1 to 10.10.10.6 (10.10.10.0 is reserved for the subnet, address 10.10.10.7 for broadcasting).

See e.g. http://www.subnet-calculator.com/subnet.php?net_class=A

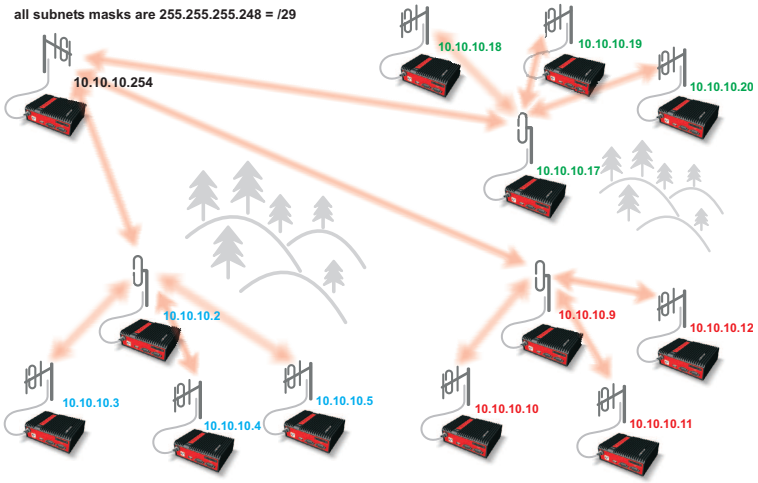


Fig. 1.2: Network with subnets

- For 10.10.10.254

```
Destination via Gateway
10.10.10.1/29  via 10.10.10.2
10.10.10.8/29  via 10.10.10.9
10.10.10.16/29 via 10.10.10.17
```

- For 10.10.10.2

```
10.10.10.8/29  via 10.10.10.9
10.10.10.16/29 via 10.10.10.17
```

- For 10.10.10.3 and 10.10.10.4 and 10.10.10.5

```
10.10.10.248/29 via 10.10.10.2
10.10.10.8/29   via 10.10.10.2
10.10.10.16/29  via 10.10.10.2
```

- For 10.10.10.9

```
10.10.10.1/29 via 10.10.10.2
10.10.10.8/29 via 10.10.10.9
```

- For 10.10.10.10 and 10.10.10.11 and 10.10.10.12

```
10.10.10.248/29 via 10.10.10.9
10.10.10.1/29   via 10.10.10.9
10.10.10.16/29  via 10.10.10.9
```

- For 10.10.10.17

```
10.10.10.1/29  via 10.10.10.2
10.10.10.16/29 via 10.10.10.17
```

- For 10.10.10.18 and 10.10.10.19 and 10.10.10.20

```
10.10.10.248/29 via 10.10.10.17
10.10.10.1/29   via 10.10.10.17
10.10.10.8/29   via 10.10.10.17
```

1.2. End devices connected over Ethernet

Both radio modem's network interfaces must be used for routing. Radio modem routing works the same as standard IP routing – for more information refer to http://www.comptechdoc.org/independent/networking/guide/netguide.pdf chapter Network Routing

Limitations:

A. **If you can set the IP address, network mask, gateway and routing table in the IP device connected to RipEX**

There are no limitations to setting up routing in this case. The only rule is that the range of radio and Ethernet IP addresses must not overlap.

B. **If you can only set the IP address, network mask and gateway, not the routing table in the IP device connected to RipEX**

In this case destination addresses must not be on the same network (i.e. the destination address must always be outside of the network mask). A destination address is the IP address of one of the devices connected to RipEX's which mutually communicate over the radio channel.

C. **If the connected device allows neither network mask, nor gateway to be set up**

Router mode cannot be used at all; use Bridge mode instead.

1.3. Ethernet addressing

If you can set up IP addresses of the end devices connected over Ethernet, you can simplify routing by hierarchic division into subnets, either complete or for routing purposes only. An example of such network layout follows.

The centre and main repeater form distinct networks with mask 255.255.255.0 (/24), the sub-networks narrow down towards the end devices 255.255.255.192 (/26) and then 255.255.255.248 (/29). Routing tables are only given for a single branch of the network for clarity. They will be similar for other RipEX's. Only Master – Slave type applications are presumed – without any direct communication between Slave devices.

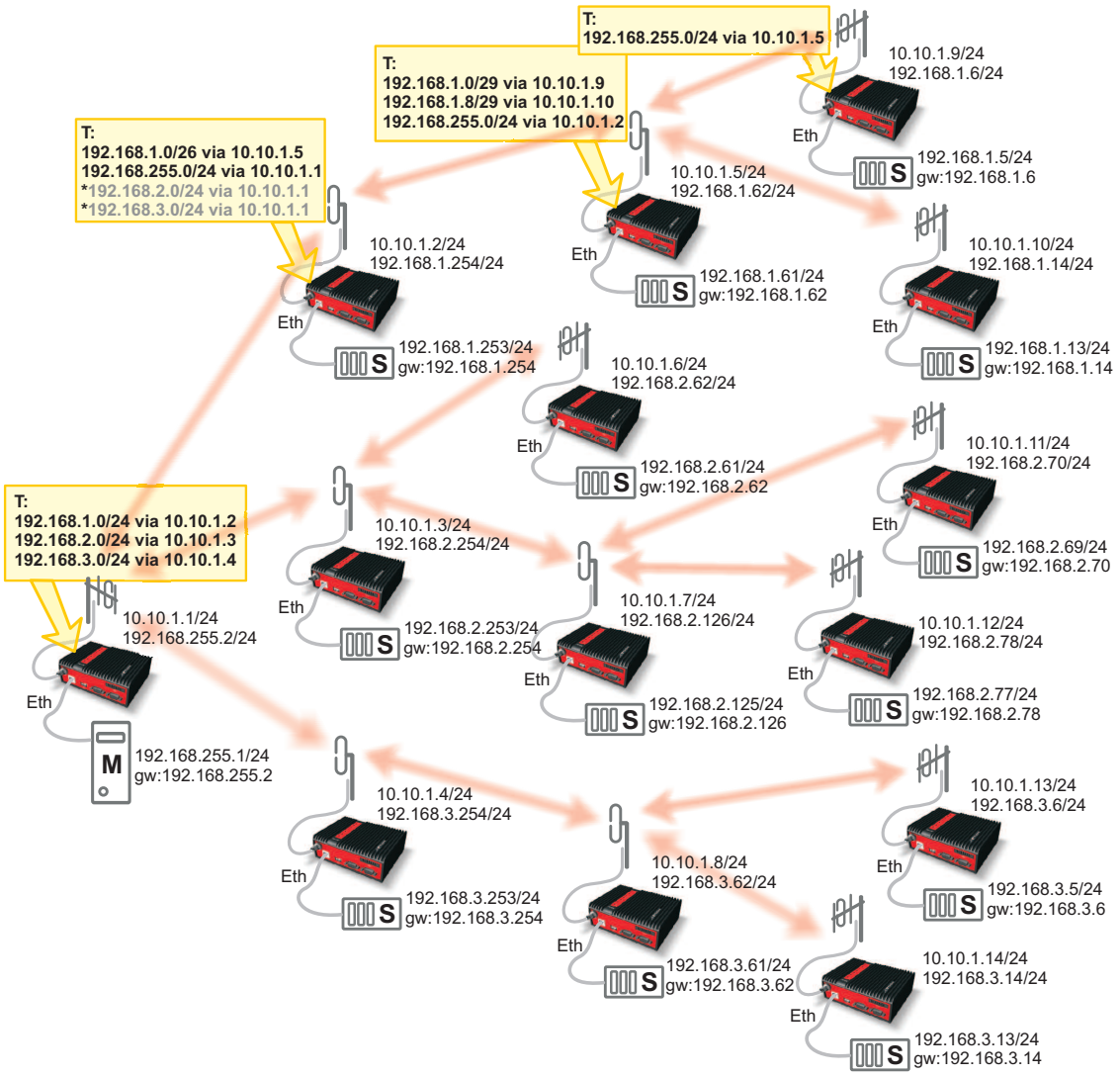


Fig. 1.3: Network with standard masks

Virtual network narrowing may also be used, while in reality narrower masks will be only used for routing purposes. This would allow you to use even the addresses reserved for network and broadcasting, though we do not recommend doing so.

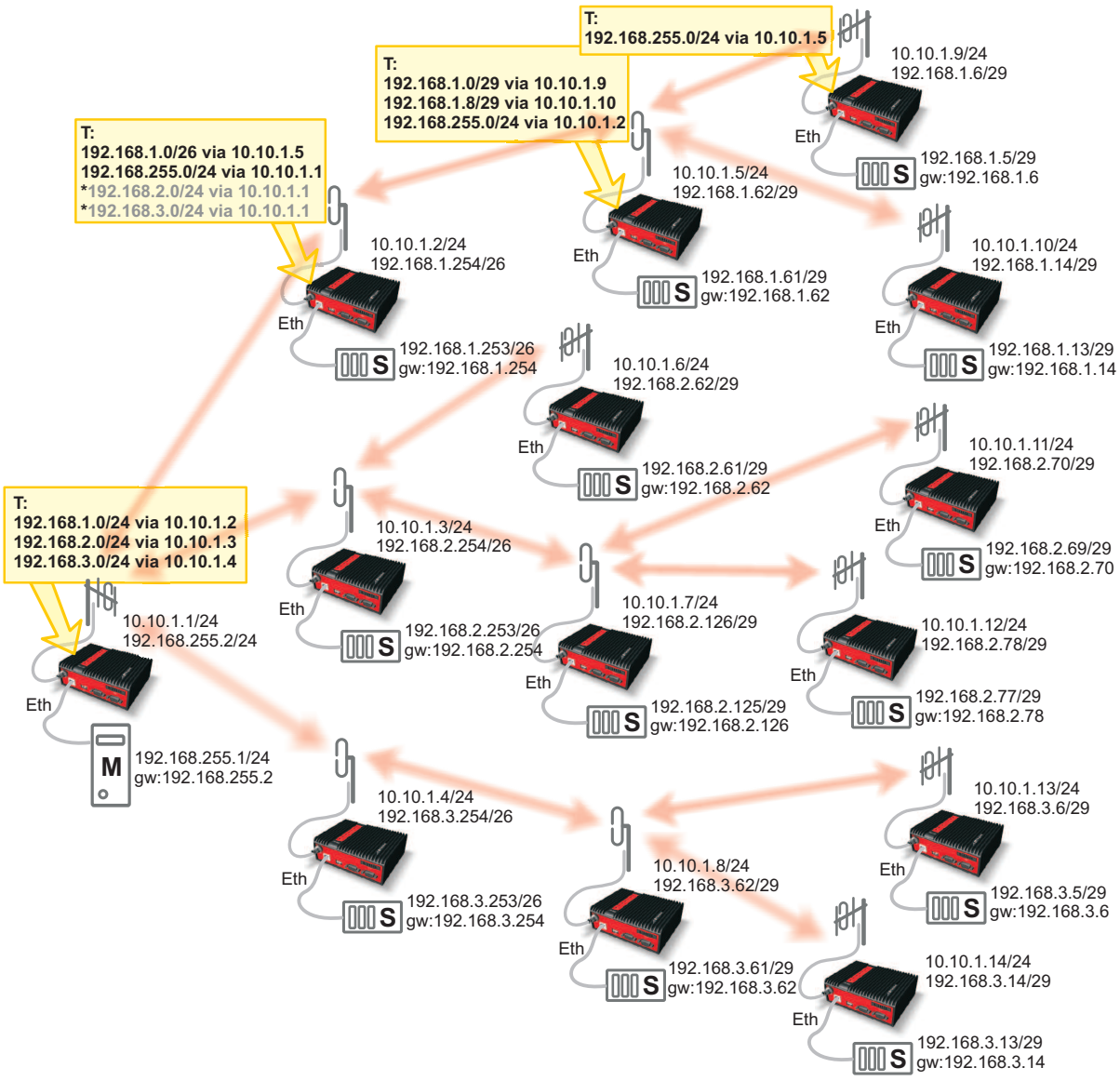


Fig. 1.4: Network with narrowed masks

2. SNMP

2.1. Simple Network Management Protocol

SNMP is a simple, widely used and useful standardised protocol typically used by Network Management Software (NMS) to read values from devices. Values can be obtained at regular intervals or on requests, saved to a database and then displayed as graphs or tables.

SNMP also enables devices to generate (trigger) the alarms by themselves and notify the NMS explicitly (SNMP traps).

2.1.1. How does SNMP work?

SNMP requires two parties for communication:

1. *SNMP "manager"* (software installed at your computer)
    - You can use commercial software or free software such as Zabbix, Zenoss, Nagios, Cacti, etc. If you want to read values manually, you can use tools such as snmpwalk, snmpget or Mib-browser software.
2. *SNMP "agent"* (a part of firmware in remote devices such as RipEX)
    - The agent receives SNMP requests to query information and responds to the manager. Several managers may read values at once and they can send their requests at any time. Alternatively, the agent sends SNMP traps whenever the monitored values (watched values in RipEX, e.g. temperature) are outside the threshold range. RipEX is capable of sending SNMP traps to up to two SNMP managers (since the firmware release 1.3).

2.1.2. SNMP communication

In SNMP, each value is uniquely identified using Object Identifier (OID). Standard communication starts by sending a request and then the response is returned. Alternatively, an agent can send an SNMP trap.

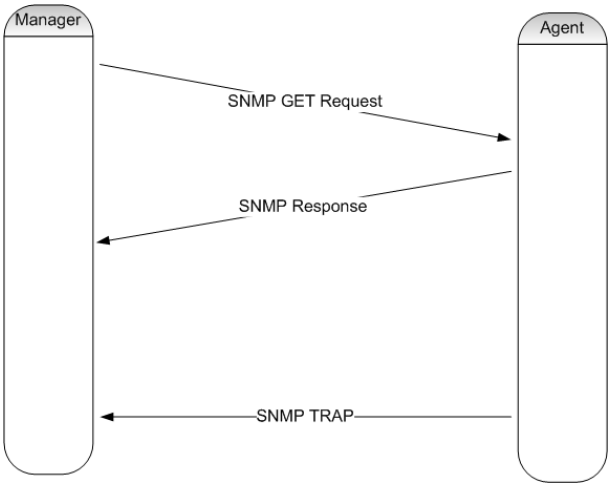


Fig. 2.1: SNMP Communication

A **request** is sent	the manager sets message-type to GET, includes OID for the required value and sets this value to NULL.
A **response** is returned	the agent sets message-type to RESPONSE and sends the requested value along with its OID back to the manager.
A **trap** is sent	to the manager without its request.

Basic Message Types

GetRequest	returns a single value.
GetNextRequest	returns the next value (using the next OID).
GetBulkRequest	returns several values in a single packet (for example, temperature, voltage, number of transmitted messages or bytes per second, etc.).
Trap	sent from the agent to the manager whenever any monitored value is beyond its thresholds.
SetRequest	used to set various parameters (unsupported by RipEX).

2.1.3. MIB database – Management Information Base

The MIB is a virtual database used for managing the entities in a communications network.

The MIB hierarchy can be depicted as a tree with a nameless root, the levels of which are assigned by different organizations. “Higher-level” MIB OIDs belong to different standards organizations, while “lower-level” OIDs are allocated by associated organizations (e.g. RACOM).

OID example:

```
RIPEX::serialNumber
serialNumber OBJECT-TYPE
-- FROM RIPEX
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Product serial number."
::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) racom(33555) ripex(2) ▶
station(1) device(1) 4 }
```

As you can see, numbers 1.3.6.1.4.1.33555 are the “higher-level” OIDs. The “lower-level” OIDs are .2.1.1.4 which are allocated by RACOM.

2.2. SNMP in RipEX

In RipEX SNMP protocol can be used to:

- Read configuration parameters from MIB,
- Read operation statistics on the radio channel, and
- Sends traps when set thresholds for monitored values are exceeded (TxLost [%], UCC, Temp, PWR, ...)

For detailed description of individual values refer to section RipEX MIB bellow.

RipEX utilises SNMP versions **SNMPv1** and **SNMPv2c** – it uses **community string** for authentication, which is by default "**public**", but can be changed. SNMP uses UDP protocol for communication; delivery checks are implemented from version 2.

Note

The RipEX MIB module complies with a validity check for severity level 3.

By default RipEX uses UDP port 161 (SNMP) for queries. Manager, which sends the query, dynamically chooses a port from which it sends its query to RipEX port 161. RipEX replies from port 161 to the dynamically selected port of the manager.

RipEX launches SNMP agent automatically on start-up if enabled. RipEX also sends alarm states (traps) to manager from port 162 (SNMPTRAP). Users can change this port number in RipEX. Traps' behaviour can be influenced (see Alarm management settings, RipEX manual, Adv. config.[1]).

When using SNMP over radio channel we recommend setting RipEX to router mode. From the point of radio network, SNMP is typically a standalone application sharing the radio channel with others. Thus it causes collisions, which are automatically resolved by the radio channel protocol in router mode. The radio channel uses no protocol in bridge mode, meaning two competing applications can only be run at a great risk of collisions and with the knowledge that packets from both applications may be irretrievably lost.

2.2.1. Limitations

SNMP is primarily designed for Ethernet networks, where generally, bandwidth capacity is not an issue. By contrast, radio bandwidth capacity is very limited and RipEX mostly works over the radio channel. For this reason, special care is recommended while configuring NMS. If badly configured, NMS can take a significant portion of the network capacity or can even overload the network completely.

Bandwidth Consumption

It is important to realise that the average size of a single request and response to a specific OID is approximately 184 Bytes each. The entire MIB for a single RipEX with one neighbouring RipEX is approximately **48 kilobytes**. Based on the limitations and the MIB size, we recommend to query only carefully selected OIDs over the radio channel and not all possible data. Set SNMP query time intervals in your NMS as long as possible. The shortest recommended interval ranges from several minutes to tens of minutes.

Wherever possible, use the RipEX Ethernet interface for SNMP communication to free up the radio channel.

Note

There are many Network Management Systems available on the market. Whichever you choose, keep in mind the described limitations. E.g. never use NMS, which can download only the entire remote device MIB and not single OIDs.

[1] http://www.racom.eu/eng/products/m/ripex/h-menu.html

Bandwidth Efficiency Tip

If you wish to monitor many watched values (VSWR, Temperature, UCC, …) from remote stations connected over the radio channel and you have a star topology network, you can improve bandwidth efficiency by reading OID values only from the Master (Repeater) RipEX station.

The advantage of the above is that the watched values from remote stations are broadcast in regular intervals and saved in the Master (Repeater) RipEX. These values from neighbouring stations have their own OID's and can be downloaded from the Master (Repeater) RipEX.

In the picture below – Master RipEX station periodically reads watched values from its neighbouring Slave stations. Whenever the NMS requests any value mentioned, the reply is sent only from the Master station (over Ethernet) saving radio bandwidth. SNMP uses radio link only for sending SNMP Traps from any Slave to the NMS.

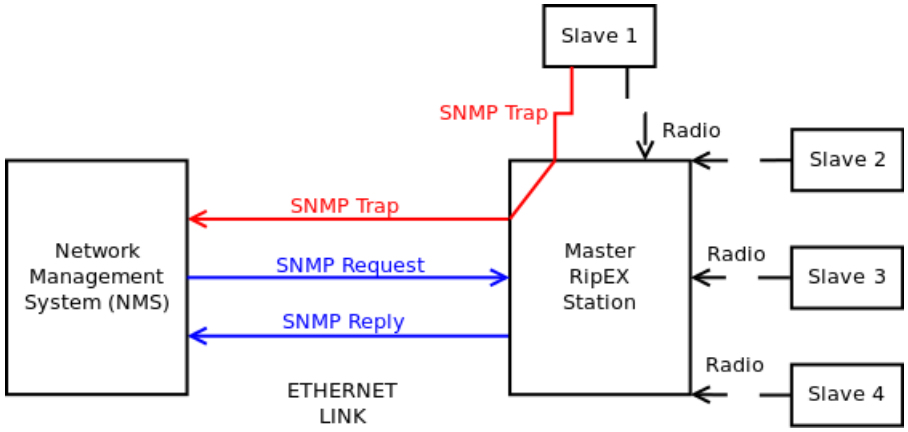


Fig. 2.2: NMS communication with Slave stations

Note

In such a case, watched values from neighbouring stations are displayed as part of the Master (Repeater) station.

2.2.2. RipEX SNMP Settings

SNMP agent is switched off by default. To enable it, go to the settings menu and click on the SNMP button. You can set the Community, turn on SNMP traps and define two trap destination IP addresses and ports.

Important

Thresholds for all SNMP traps can be configured in the RipEX web interface, Settings → Alarm management. Since detailed description of RipEX SNMP traps settings can vary based on the current firmware, please kindly refer to the online Help accessible through the RipEX web interface or see the User manual, Chapter Settings (http://www.ra-com.eu/eng/products/m/ripex/h-menu.html#settings).

2.2.3. RipEX Traps Description

The trap is sent whenever any of the following watched values are beyond their threshold ranges:

- RSS (Received Signal Strength)
- DQ (Data Quality)
- TX Lost (The probability of a transmitted frame being lost)
- UCC (Power voltage [V])
- Temperature [C]
- RF Power [W]
- VSWR (Voltage Standing Wave Ratio, 1.0 = the best ratio, 1.0 – 1.8 = acceptable ratio, > 2.5 = indicates a serious problem in antenna or feeder)
- Ethernet RX/TX Packets ratio (Ratio between received and sent packets over Ethernet)
- COM1/2 RX/TX Packets ratio (Ratio between received and sent packets over COM ports)
- HW Alarm input
- Hot-Standby (SNMP trap containing active station identity – sent by the active station)
- Backup paths system (Backup path state and Alternative path state changes)
- Unit ready (the hardware alarm output or the SNMP trap indicates that the RipEX radio is ready to operate)

2.3. Network Management System – ZABBIX

To access our SNMP values, any Network Management System (NMS) can be used. However, we recommend using the ZABBIX open source monitoring system. It can be downloaded at: http://www.zabbix.com/download.php.

Zabbix website provides the following short description:
Zabbix is the ultimate open source availability and performance monitoring solution. Zabbix offers advanced monitoring, alerting, and visualization features today which are missing in other monitoring systems, even some of the best commercial ones.

If you have chosen the Zabbix software, please read the following pages where we offer a basic Starting Guide to RipEX and Zabbix co-working. The sections may in any case be helpful for some generally applicable hints and tips.

Note

The following guide was tested with Zabbix release 2.2.0. It should work on any 2.0.x release or any newer 2.2.x release. If you use any of the 1.8.x releases, some tasks may need a different approach.

Take the opportunity to remotely access and test a live Zabbix demo[2]. Contact us for access[3] details.

2.3.1. Installation and Documentation

Due to security requirements and mission-critical nature of monitoring server, UNIX is the only operating system that can consistently deliver the necessary performance, fault tolerance and resilience.

Zabbix is tested on the following platforms:

- Linux
- IBM AIX
- FreeBSD
- NetBSD

[2] http://www.racom.eu/eng/products/m/ripex/demo/zabbix.html
[3] http://www.racom.eu/eng/products/remote-access.html#load(product=zabbix)

- OpenBSD
- HP-UX
- Mac OS X
- Solaris
- Windows: 2000, Server 2003, XP, Vista, Server 2008, 7 (Zabbix agent only)

For further details, visit Zabbix Documentation at http://www.zabbix.com/documentation.php. It contains a large body of information about installation steps, configuration, performance etc. If you are unsure how to proceed with any task, refer to the Zabbix documentation first. You can find an installation guide there, too.

This Guide does not present all Zabbix settings, but should help you incorporate the RipEX SNMP functionality into the Zabbix software.

Note

The following guide requires using MySQL database used in Zabbix. If you choose another one, you will need to alter at least the trap handling bash script provided.

Windows Installation

If you need to use Windows platform as the host operating system for Zabbix, you can install VMware/VirtualBox software and install Zabbix Appliance. The Zabbix Appliance can be downloaded from http://www.zabbix.com/download.php. Please remember that Zabbix Appliance is not intended for serious production use at this time.

VMware download: https://www.vmware.com/support/
VirtualBox download: https://www.virtualbox.org/wiki/Downloads
See the respective documentation on how to install and use virtualisation software.

2.3.2. Templates

After successful installation, you can import any of the predefined templates. Each template is the collection of Zabbix Items corresponding to a set of OIDs, triggers, graphs and applications. The template can be easily linked to any monitored host (RipEX) and you can have access to the desired values very quickly.

What Templates do we Provide?

The Templates list:

- Name: RipEX Template
  - Consists of all specific OIDs provided by RACOM
  - Implements one neighbouring RipEX monitoring
  - 17 Applications, 237 Items, 4 Triggers, 5 Graphs
- Name: RipEX – RFC1213 Template
  - Consists of supported RFC1213 OIDs
  - 1 Application, 56 Items
- Name: RipEX – RS232 Template
  - Consists of supported RS232 OIDs
  - 1 Application, 21 Items
- Name: RipEX – SNMP Trapper Template
  - Consists of SNMP trapper items, which are triggered by 15 kinds of traps

  - 1 Application, 15 Items, 15 Triggers
- Name: PING Template
  - Pings a defined host and triggers whenever the host is unreachable
  - 1 Application, 2 Items, 1 Trigger

All templates can be downloaded from the RipEX Download site at http://www.racom.eu/download/hw/ripex/free/eng/3_fw/RipEX_Zabbix_templ.zip.

How do I Import the RipEX Templates?

In order to import the template, click on the **Configuration** → **Templates** button at the top of the Zabbix web page. At top right corner you will see the **Import Template** button, click on it.

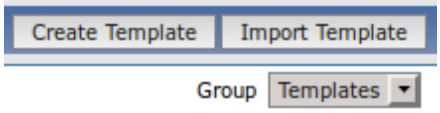


Fig. 2.3: Importing Template button

Important

Before importing the template file, see Chapter 2.3.4 Value Mappings. Since the Zabbix release 2.2.1, it is mandatory to define the value mappings before importing the required templates.

Select the RipEX template file and Import that file. Repeat this step for each template.

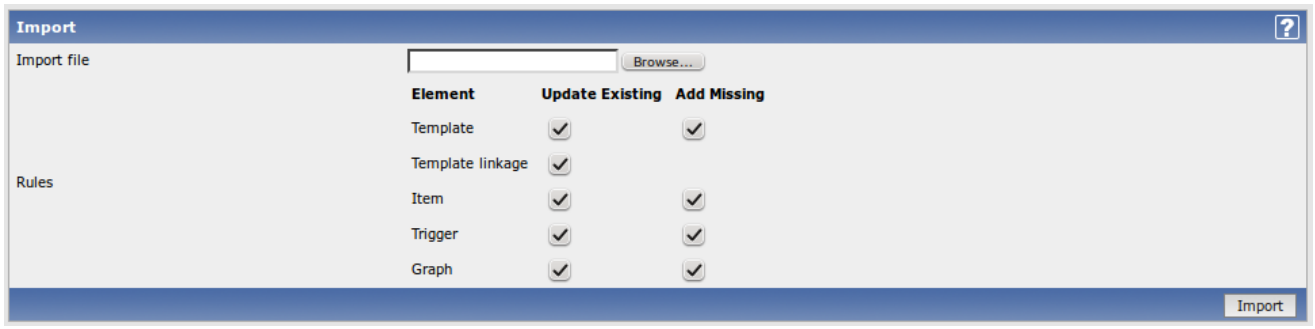


Fig. 2.4: Importing Template options

Now you can see all RipEX templates at the Template list window among all other templates which are in Zabbix by default.

Note

If you already imported the template and you need to update it, just import the newer version with the same name and the current template will be automatically updated.

Each **Item** has its Description, SNMP OID number, community string, UDP port (161), key, update interval and other parameters. One of the key parameter is the update interval, because it defines how often Zabbix will request various replies from the RipEX stations. This interval is predefined to 30 minutes, but you should consider changing it to suit your radio network infrastructure.

The individual items can be in an active or disabled state. By default, only some items are active based on their importance – see the next chapter for more information. If you wish to monitor more values, activate the desired ones. But as already mentioned, preferably use the RipEX Ethernet interface for SNMP communication to free up the radio channel. If this is not possible, consider carefully whether monitoring other values is necessary.

Always monitor only the values which you really need and with reasonable update times.

The items are divided into the usage groups, called **Applications** in Zabbix. These applications serve for better clarification of the defined items.

If you wish to be notified whenever any monitored value is out of its threshold range, you can define a **Trigger** for it. These notifications are viewable on the Zabbix dashboard, item history or you can have e-mail/jabber/sms notifications enabled. Each notification can have one of six predefined severity levels (warning, critical, …).
We also provide several triggers within the templates. Triggers defined in templates cannot be edited within individual hosts, which means you cannot define various threshold ranges for hosts and each host would have the same threshold range. Please define your own triggers within each individual host.

Note

You can use a Clone option to create a copy of any template item or trigger for an individual host. In this case, you can edit its predefined values to meet your requirements for each host separately.

Graphs are automatically created for each monitored numeric value, but you can also create special graphs with several values on a single graph. We provide 5 predefined graphs containing some basic watched values like temperature, UCC etc.

For more information, see the Zabbix documentation. You can delete, add or edit any template component. The predefined state serves as a quick start, but you do not have to use them at all and you can create your own set of monitored values/items.

Which Values/Items Should I Monitor?

The templates themselves are fully scalable and consist of many items. However, monitoring all of them is not required in a routine situation. Pre-activated items in RipEX default templates are:

- RipEX Template
  - Pre-activated Items: 7
  - Default Update Time: 30 minutes
    - Local Station:
      - Modem temperature (°C), RF power (W), TX lost (%), UCC (V), VSWR
    - Remote Station:
      - DQ, RSS (dBm)
- RipEX – RFC1213 Template
  - All items are disabled by default
- RipEX – RS232 Template
  - All items are disabled by default
- RipEX – SNMP Trapper Template
  - All SNMP trap items and triggers are enabled by default except DQ and RSS. These triggers need to be cloned for individual hosts, because we cannot predefine remote hosts IP addresses. See Chapter 2.4.2 for more information.
- PING Template

- Pre-activated Items: 1
- Default Update Time: 30 minutes
- The only active item checks the host reachability and triggers an alarm if the host is not reachable.

Note

If you need to monitor more than one remote RipEX station, you need to “clone” existing items for the remote station watched values.

Configuration example of **the second remote station** (Data Quality item):

- Go to the **Template list window** and click on the items of *RipEX template*.
- Find the item called *“Remote station 1 wv – Average DQ value”* and click on it.
- Find the item called *“Remote station 1 wv – Average DQ value”* and click on it.
- Change the *“Key”* parameter to have “*2*” in the brackets – *wvRemDqAvg[2*].
- Change the last digit of the SNMP OID to be "2" instead of "1" (i.e. 1.3.6.1.4.1.33555.2.4.3.1.7.**2**).
- Click on the “*Clone*” button.
- Click on the “*Save*” button.
- Now every station using this template will monitor two remote stations for their DQ value.
- If you want to define that item for individual RipEX stations only, you need to clone the item within the **host** configuration *Item list*. To enable the settings to change, you need to click on the *“Clone”* button first; the software will then allow you to change the settings.

Total bandwidth used (requests, replies) – if monitoring only the values defined above:

- Approx. 3 kB / 1 RipEX station / 1 hour

2.3.3. How to Import Monitored RipEX Stations?

Now you have a working template, but you need to define hosts (RipEX stations). Each RipEX station has its own IP address. The following steps will guide you through the Host Configuration.

To create a host, go to **Configuration** → **Hosts** and click on the **Create Host** button. Define the Host name and its IP address.

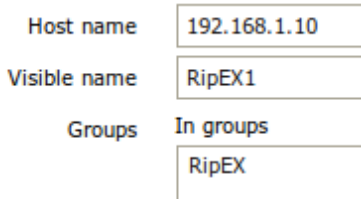


Fig. 2.5: Defining the Host name and its IP address

You can optionally define a Group for the hosts. Creating a **Group** is straightforward. You can create a new one while creating a host or you can do so by going to the **Configuration** → **Groups** tab and clicking on the **Create Group** button.

Linking a template to the host(s) can be achieved under the same tab or you can open Template settings and link any desired host to it.

You have to set the IP address and port number for the SNMP interface (port 161). Otherwise, you won't be able to use any SNMP item.

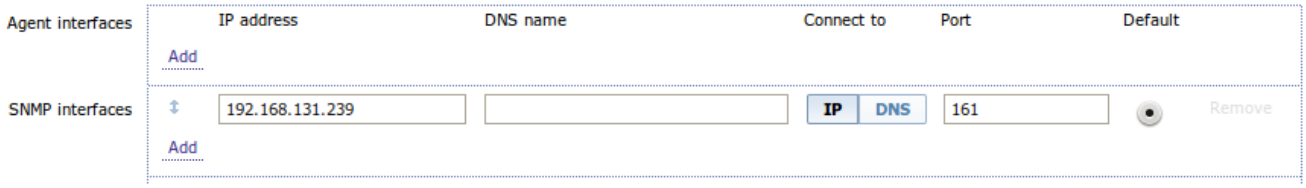


Fig. 2.6: Defining the SNMP interface

Where can I See the RipEX Monitored Values?

To check monitored values, go to the **Monitoring** → **Latest data** tab and choose the desired host from the Menu.

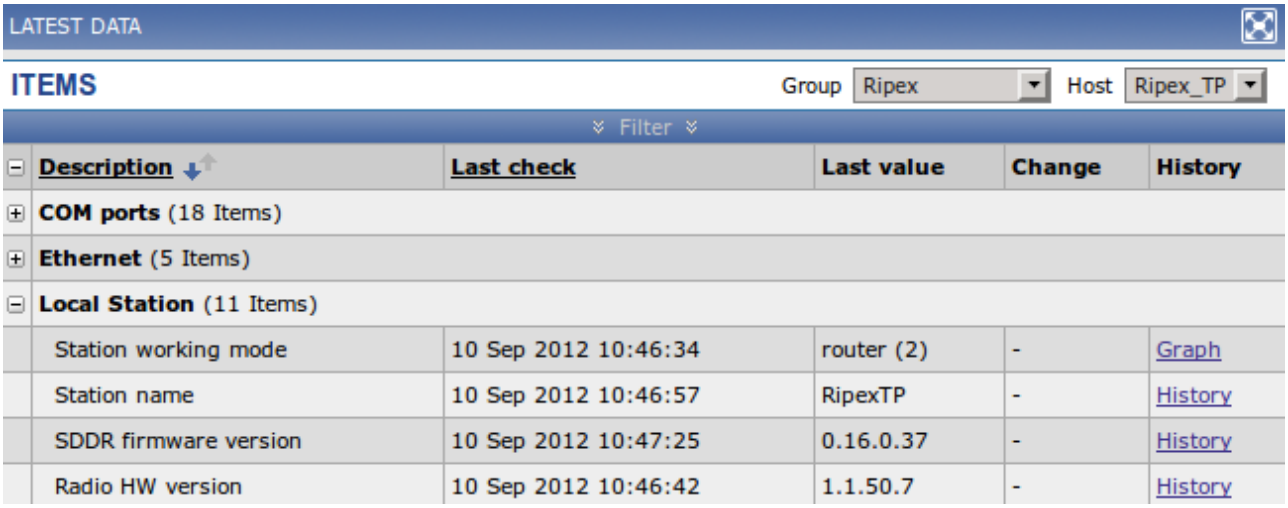


Fig. 2.7: RipEX latest data

For each item, you can see a graph or a history table. If a trigger is configured for the item, the graph shows a line with a threshold value.

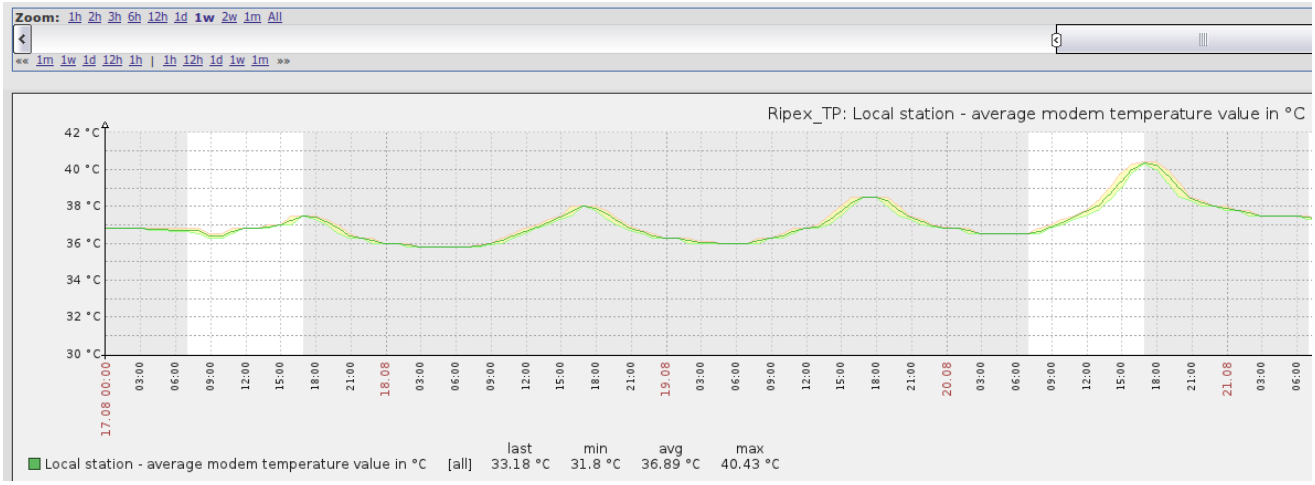

Zoom: 1h 2h 3h 6h 12h 1d 1w 2w 1m All
Ripex_TP: Local station - average modem temperature value in °C
42 °C
40 °C
38 °C
36 °C
34 °C
32 °C
30 °C
Local station - average modem temperature value in °C [all] last 33.18 °C min 31.8 °C avg 36.89 °C max 40.43 °C

Fig. 2.8: RipEX graph

2.3.4. Value Mappings

Responses of Several OID objects are unsigned integers, but these values have a special meaning.

Example 2.1. deviceMode

- “1” stands for “bridge” mode.
- “2” stands for “router” mode.

Unfortunately, by default, you can see only the numeric values at the Zabbix front-end. You need to create the value mappings for all OID objects of this kind manually. The value mapping is not exported within the RipEX template.

Create the value mappings before importing the RipEX template. Otherwise you will need to link all value mappings with the appropriate items manually.

Remember that since Zabbix version 2.2.1, it is mandatory to create the value mappings before importing the templates.

Note

This syntax feature is used throughout all MIB tables, not only the RipEX MIB table.

To add new value mappings, go to *Administration* → *General* → *Value Mapping*. Click on the “**Create value map**” button and insert the values, which are mentioned on the following lines. There is an Item list, which uses these value mappings (either link them manually or automatically by importing the template).

Note

There are also several value mappings used at RFC1213 and RS232.

Value Mappings List

RipEX.AlmState	**Items:**
-1 ⇒ unknown	Alarm state - COM1 interface Rx to Tx packets ratio
0 ⇒ inactive	Alarm state - COM2 interface Rx to Tx packets ratio
1 ⇒ active	Alarm state - Device temperature
	Alarm state - DQ
	Alarm state - ETH interface Rx to Tx packets ratio
	Alarm state - HW Input
	Alarm state - RF Power
	Alarm state - RSS
	Alarm state - Tx lost
	Alarm state - UCC
	Alarm state - Unit ready
	Alarm state - VSWR

RipEX.BackupPathsState	**Items:**
0 ⇒ unknown	Backup Paths 1 - Alternative Paths - Currently passive paths State
1 ⇒ up	Backup Paths 2 - Alternative Paths - Currently passive paths State
2 ⇒ down	Backup Paths 1 - Alternative Paths - Currently used path State
	Backup Paths 2 - Alternative Paths - Currently used path State

RipEX.comProtocol:	**Items:**
0 ⇒ none	COM1 - Protocol
3 ⇒ AsyncLink	COM2 - Protocol
4 ⇒ Modbus	TS 1 COM user protocol type
5 ⇒ IEC101	TS 2 COM user protocol type
6 ⇒ DNP3	TS 3 COM user protocol type
7 ⇒ UNI	TS 4 COM user protocol type
8 ⇒ Comli	TS 5 COM user protocol type
9 ⇒ DF1	
10 ⇒ Profibus	
12 ⇒ C24	
13 ⇒ RP570	
14 ⇒ Cactus	
15 ⇒ ITT Flygt	
16 ⇒ SLIP	
16 ⇒ Siemens 3964 (R)	

RipEX.deviceMode	**Items:**
1 ⇒ bridge	Station working mode
2 ⇒ router	

RipEX.eDhcp	**Items:**
0 ⇒ off	Ethernet interface DHCP mode
1 ⇒ server	
2 ⇒ client	

RipEX.eSpeed	**Items:**
0 ⇒ auto	Ethernet interface bit rate and duplex settings
1 ⇒ s-100baseTX-Full	
2 ⇒ s-100baseTX-Half	
3 ⇒ s-10baseT-Full	
4 ⇒ s-10baseT-Half	

RipEX.ifTmATM	**Items:**
0 ⇒ mask	TCP Modbus COM protocol address translation mode
1 ⇒ table	

RipEX.IOState	**Items:**
-1 ⇒ unknown	HW alarm input contact state
0 ⇒ off	
1 ⇒ on	

RipEX.RelayContactType	**Items:**
0 ⇒ off	HW alarm input contact type
1 ⇒ normally-closed	
2 ⇒ normally-open	

RipEX.rEncryption	**Items:**
0 ⇒ off	Radio interface encryption method
1 ⇒ aes256	

RipEX.rRfPwr	**Items:**
0 ⇒ mE-100mW	Radio interface RF power
1 ⇒ mEr-200mW	
2 ⇒ mE-500mW	
3 ⇒ mE-1W	
4 ⇒ mE-2W	
5 ⇒ mE-3W	
6 ⇒ mE-4W	
7 ⇒ mE-5W	
8 ⇒ mE-10W	
9 ⇒ mE-8W	
17 ⇒ mL-200W	
18 ⇒ mL-500mW	
19 ⇒ mL-1W	
20 ⇒ mL-2W	

RipEX.SettingState	**Items:**
0 ⇒ off	Ethernet interface broadcast and multicast status
1 ⇒ on	Ethernet interface shaping status
	Terminal server status
	TCP Modbus COM protocol broadcast accept
	Radio interface FEC
	TS 1 on/off
	TS 2 on/off

RipEX.SettingState

Items:

TS 3 on/off

TS 4 on/off

TS 5 on/off

RipEX.tsEthProtType

0 ⇒ tcp

1 ⇒ udp

Items:

TS 1 Ethernet protocol type

TS 2 Ethernet protocol type

TS 3 Ethernet protocol type

TS 4 Ethernet protocol type

TS 5 Ethernet protocol type

RFC1213.ifType

1 ⇒ other

2 ⇒ regular1822

3 ⇒ hdh1822

4 ⇒ ddn-x25

5 ⇒ rfc877-x25

6 ⇒ ethernet-csmacd

7 ⇒ iso88023-csmacd

8 ⇒ iso88024-tokenBus

9 ⇒ iso88025-tokenRing

10 ⇒ iso88026-man

11 ⇒ starLan

12 ⇒ proteon-10Mbit

13 ⇒ proteon-80Mbit

14 ⇒ hyperchannel

15 ⇒ fddi

16 ⇒ lapb

17 ⇒ sdlc

18 ⇒ ds1

19 ⇒ e1

20 ⇒ basicISDN

21 ⇒ primaryISDN

22 ⇒ propPointToPointSerial

23 ⇒ ppp

24 ⇒ softwareLoopback

25 ⇒ eon

26 ⇒ ethernet-3Mbit

Items:

RFC1213 - Interface 1 - The type of interface (physical/link protocol)

RFC1213 - Interface 2 - The type of interface (physical/link protocol)

RFC1213.ifType	**Items:**
27 ⇒ nsip	
28 ⇒ slip	
29 ⇒ ultra	
30 ⇒ ds3	
31 ⇒ sip	
32 ⇒ frame-relay	

RFC1213.ipForwarding	**Items:**
1 ⇒ forwarding	RFC1213 - The indication of whether this entity is acting as an IP gateway
2 ⇒ not-forwarding	

RFC1213.snmpEnableAuthenTraps	**Items:**
1 ⇒ enabled	RFC1213 - SNMP - Indicates whether the SNMP agent process is permitted to generate authentication-failure traps
2 ⇒ disabled	

RS232.rs232AsyncPortParity	**Items:**
1 ⇒ none	RS232 port 1 - The port's sense of a character parity bit
2 ⇒ odd	RS232 port 2 - The port's sense of a character parity bit
3 ⇒ even	
4 ⇒ mark	
5 ⇒ space	

RS232.rs232AsyncPortStopBits	**Items:**
1 ⇒ one	RS232 port 1 - The port's number of stop bits
2 ⇒ two	RS232 port 2 - The port's number of stop bits
3 ⇒ oneAndHalf	
4 ⇒ dynamic	

RS232.rs232PortInFlowType	**Items:**
1 ⇒ none	RS232 port 1 - The port's type of input flow control
2 ⇒ ctsRts	RS232 port 2 - The port's type of input flow control
3 ⇒ dsrDtr	RS232 port 1 - The port's type of output flow control
	RS232 port 2 - The port's type of output flow control

RS232.rs232PortType	**Items:**
1 ⇒ other	RS232 port 1 - The port's hardware type
2 ⇒ rs232	RS232 port 2 - The port's hardware type
3 ⇒ rs422	
4 ⇒ rs423	
5 ⇒ v35	

RS232.rs232PortType	**Items:**
6 ⇒ x21	

ICMP ping - Accessibility	**Items:**
0 ⇒ ICMP ping fails	ICMP ping - Accessibility
1 ⇒ ICMP ping successful	

Note

Two value mappings should already be included in the Zabbix itself, see "SNMP interface status (ifAdminStatus)" and "SNMP interface status (ifOperStatus)" in the Value mapping menu. Four Items from the RFC1213 template use these mappings.

How can I Edit an Item to Link with a Value Map?

Go to **Configuration** → **Templates** and choose one of the imported template. Open the item configuration window and click on the chosen item to view and edit its settings.

Choose the appropriate value map in the Menu “Show value” and save the changes.

Example: RipEX.eDhpc

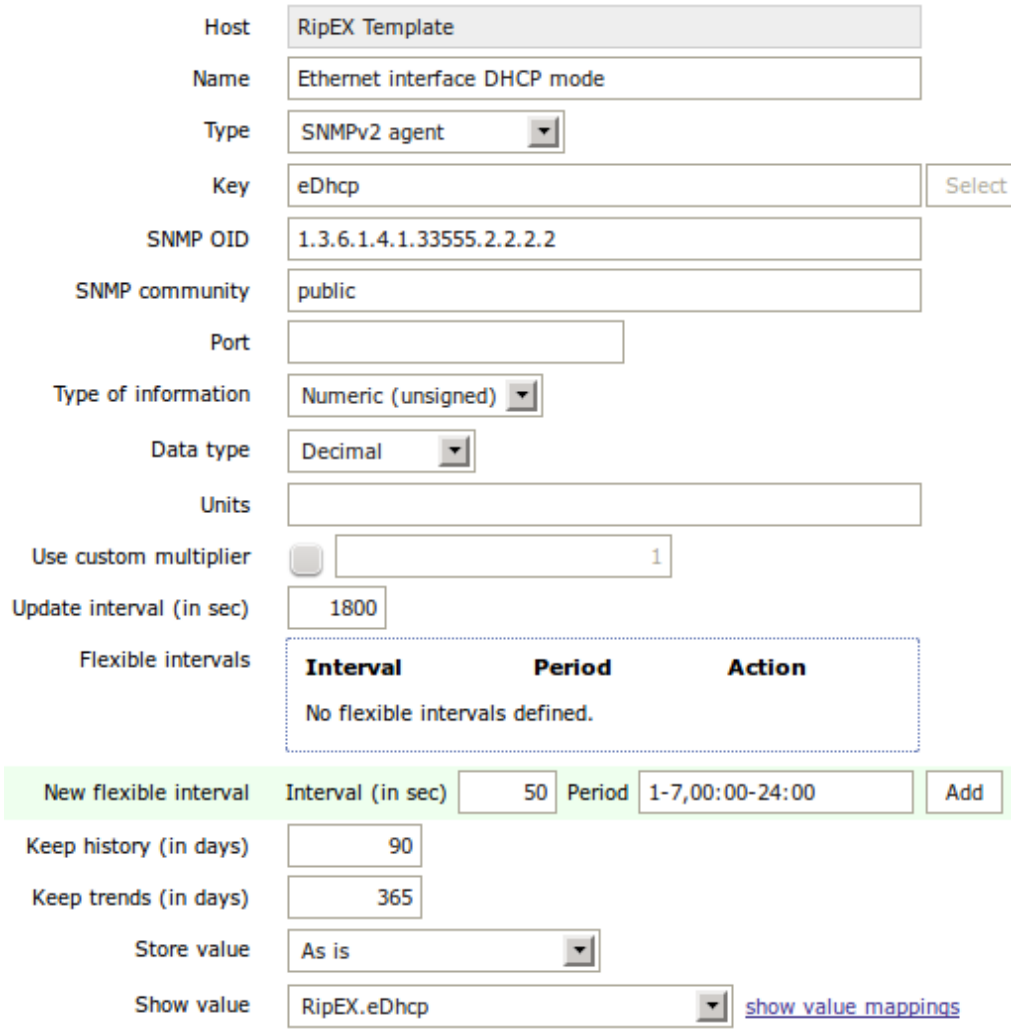


Fig. 2.9: Linking a value map to an item

2.4. How do I Know that Something Has Happened to the RipEX Station?

There are two ways to check the RipEX stations. You can actively query the station in the defined time intervals or you can just wait for the trap to be received.

2.4.1. Active Querying

This option is the one already mentioned. You have a defined item which is updated e.g. every 10 minutes. It means that every 10 minutes, Zabbix requests a reply to the SNMP GET message for the specific OID object and it stores these values in the database.

A trigger can be configured for each item. For instance, temperature threshold alarm is set to 50°C. Whenever Zabbix receives an SNMP RESPONSE message from any monitored host with temperature higher than 50°C, an alarm is triggered. If the alarm is triggered, it is displayed at the Zabbix Dashboard. The Alarm will be visible in the "**Last 20 issues**" table and you will see which host is having an issue in the "**Host status**" table.

When the temperature falls back into the allowed range, the issue will be deleted from the Zabbix dashboard.

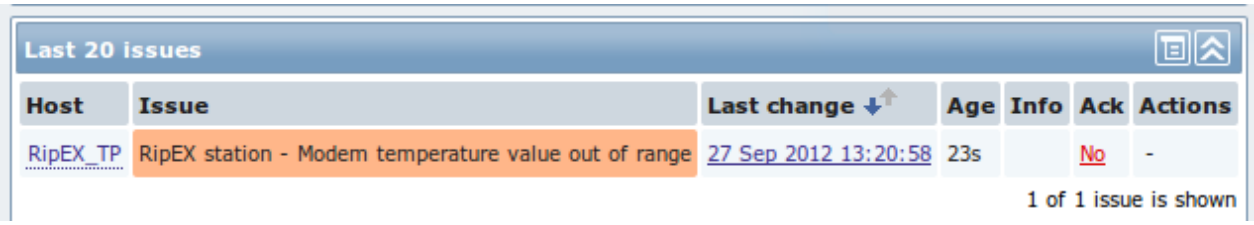


Fig. 2.10: Displaying of RipEX trap

2.4.2. SNMP Traps

The key aspect of the SNMP are the TRAPS. These OID objects are not actively monitored by the Zabbix manager but by the RipEX itself. This behaviour is described in the on-line help on RipEX web Settings page or in the User manual, Chapter 7.3.

How to Configure Traps in Zabbix?

That, unfortunately, is rather tricky. There are several ways how to configure traps – only one of them will be explained in this guide.

Note

Another approach could be using SNMPTT functionality.

You have to install an snmptrapd, a daemon which receives SNMP traps and pass them into the Zabbix front-end.

You can use the script (*snmptrap.sh*) which is included in the RipEX_Zabbix_templ.zip file downloadable from http://www.racom.eu/eng/products/radio-modem-ripex.html#download website. Copy the script file into /home/zabbix/bin/ directory and change the file privileges and make it executable.

```
# mkdir -p /home/zabbix/bin; chown zabbix /home/zabbix
# cp misc/snmptrap/snmptrapd.sh /home/zabbix/bin
# chmod +x /home/zabbix/bin/snmptrapd.sh
```

After that, you need to edit the file. By executing

```
$ which zabbix_sender
```

you will find the full path to this executable binary file. Change the path in the file, e.g.

```
ZABBIX_SENDER="/usr/bin/zabbix_sender";
```

Make sure, you have set other parameters correctly, e.g.:

```
ZABBIX_SERVER="localhost";
ZABBIX_PORT="10051";
KEY="snmptraps";
HOST="snmptraps";
```

SERVER and PORT could be different as per your configuration. KEY and HOST has default values, they will be changed depending on type of trap received.

After that, the script parses the output of each received SNMP trap, looks to the MySQL database to select appropriate host and declares an associative array containing trap descriptions. Eventually, it sends the whole message to your Zabbix server.

The script logs trap information into the /tmp/trap_messages.log file.

Note

If you have MySQL database protected by a password, you need to change this line:

```
ZABBIXHOST=$(echo "select host from zabbix.hosts where host=\"$sourceIp\" order by 'hostid' ▶
limit 1;" | mysql -N 2> /dev/null)
```

to

```
ZABBIXHOST=$(echo "select host from zabbix.hosts where host=\"$sourceIp\" order by 'hostid' ▶
limit 1;" | mysql -N -u<USER> -p<PASSWORD> 2> /dev/null)
```

Replace the <USER> and <PASSWORD> to correspond to your settings.

OK, now we have our script prepared, let's configure the Zabbix front-end:

If you have not yet done so, import the template file zbx_templates_trapper.xml. This template includes Zabbix trapper and should be linked to each monitored RipEX station. It will cause all received traps for known hosts to be mapped to the appropriate host.

Note

If Zabbix receives a trap for an unknown host it will not be displayed.

The host MUST to be configured using the IP address as the Host name, e.g.:

Host name: 192.168.10.1

Visible name: RipEX1

SNMP interface: 192.168.10.1, port 161, IP

Along with this template, 15 new items and triggers appear at each used host. That is exactly the number of SNMP traps defined at the RipEX. Each trap should be recognized and the Zabbix should display the correct information message at the dashboard.

Severity	Name	Expression	Status
Information	Alternative path state has changed within last 5 minutes	{RipEX SNMP Trapper Template:trpBpathAlt.**nodata**(300)}=0	Enabled
Information	Backup path state has changed within last 5 minutes	{RipEX SNMP Trapper Template:trpBpath.**nodata**(300)}=0	Enabled
Average	RipEX station - COM1 RX/TX packet ratio out of range	({RipEX SNMP Trapper Template:trpCom1Pr.**str**("OFF")}=0)	Enabled
Average	RipEX station - COM2 RX/TX packet ratio out of range	({RipEX SNMP Trapper Template:trpCom2Pr.**str**("OFF")}=0)	Enabled
Average	RipEX station - Ethernet Rx/Tx packet ratio out of range	({RipEX SNMP Trapper Template:trpEthPr.**str**("OFF")}=0)	Enabled
Average	RipEX station - HW input in the alarm state	({RipEX SNMP Trapper Template:trpHwIn.**str**("OFF")}=0)	Enabled
Information	RipEX station - Modem becomes active in a Hot-Standby mode within the last 5 minutes	({RipEX SNMP Trapper Template:trpHotStby.**nodata**(300)}=0)	Enabled
Average	RipEX station - Modem temperature value out of range	({RipEX SNMP Trapper Template:trpTemp.**str**("OFF")}=0)	Enabled
Average	RipEX station - Remote station (X.Y.Z.W) DQ value out of range	({RipEX SNMP Trapper Template:trpDqX.Y.Z.W.**str**("OFF")}=0)	Disabled
Average	RipEX station - Remote station (X.Y.Z.W) RSS value out of range	({RipEX SNMP Trapper Template:trpRssX.Y.Z.W.**str**("OFF")}=0)	Disabled
Average	RipEX station - RF power value out of range	({RipEX SNMP Trapper Template:trpRfpwr.**str**("OFF")}=0)	Enabled
Information	RipEX station - The unit Ready signal is UP (1)	({RipEX SNMP Trapper Template:trpUnitReady.**str**("OFF")}=0) & ({RipEX SNMP Trapper Template:trpUnitReady.**nodata**(300)}=0)	Enabled
Average	RipEX station - TX Lost value out of range	({RipEX SNMP Trapper Template:trpTxLost.**str**("OFF")}=0)	Enabled
Average	RipEX station - UCC value out of range	({RipEX SNMP Trapper Template:trpUcc.**str**("OFF")}=0)	Enabled
Average	RipEX station - VSWR value out of range	({RipEX SNMP Trapper Template:trpVswr.**str**("OFF")}=0)	Enabled

Fig. 2.11: Definition of RipEX traps

RipEX sends a trap whenever the watched value is out of range or other condition configured is met (e.g. changes in Backup paths). When the watched value returns to its configured interval, RipEX sends a trap again to notify Zabbix (or any other NMS). All triggers stay active until this notification is received.

RSS and DQ trap items are disabled in the template by default. The reason is that we need to define remote RipEX IP addresses first. See the following example for enabling a DQ trap:

Go to the Zabbix web front-end and select a RipEX host for which you want to process DQ traps. Click on the Items button and find an item with the following key: *trpDqX.Y.Z.W*.

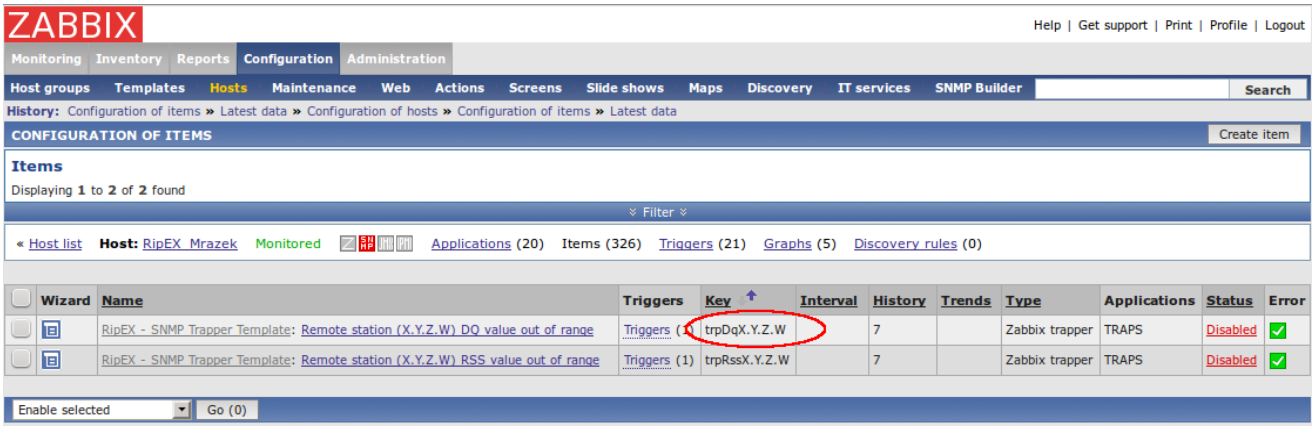

ZABBIX
Help | Get support | Print | Profile | Logout
Monitoring Inventory Reports Configuration Administration
Host groups Templates Hosts Maintenance Web Actions Screens Slide shows Maps Discovery IT services SNMP Builder
Search
History: Configuration of items » Latest data » Configuration of hosts » Configuration of items » Latest data
CONFIGURATION OF ITEMS
Create item
Items
Displaying 1 to 2 of 2 found
Filter
« Host list Host: RipEX_Mrazek Monitored Applications (20) Items (326) Triggers (21) Graphs (5) Discovery rules (0)
Wizard Name Triggers Key Interval History Trends Type Applications Status Error
RipEX - SNMP Trapper Template: Remote station (X.Y.Z.W) DQ value out of range Triggers (1) trpDqX.Y.Z.W 7 Zabbix trapper TRAPS Disabled
RipEX - SNMP Trapper Template: Remote station (X.Y.Z.W) RSS value out of range Triggers (1) trpRssX.Y.Z.W 7 Zabbix trapper TRAPS Disabled
Enable selected
Go (0)

Fig. 2.12: Default DQ trap item

Click on the item and then click on the Clone button. Now you can edit the item. Replace the "X.Y.Z.W" string in the item Name with the remote RipEX IP address (e.g. 192.168.131.55). Do the same in the Key field and select the Enabled option in the Status field. See the following example:

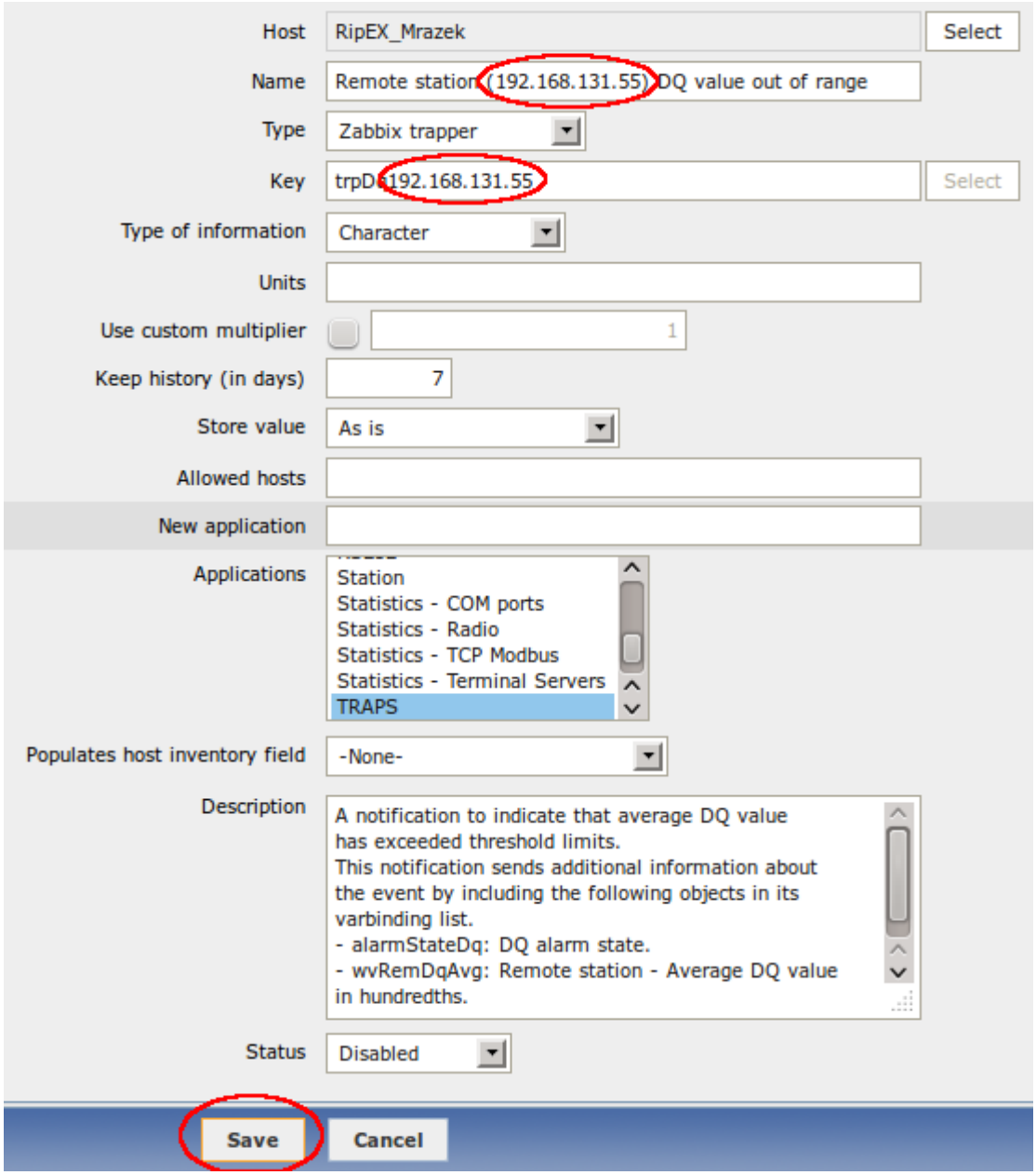


Fig. 2.13: Edited DQ trap item

Save the changes and open the host Triggers list. Repeat the above steps for the DQ trigger and save the changes. You should see the trigger with the enabled status.

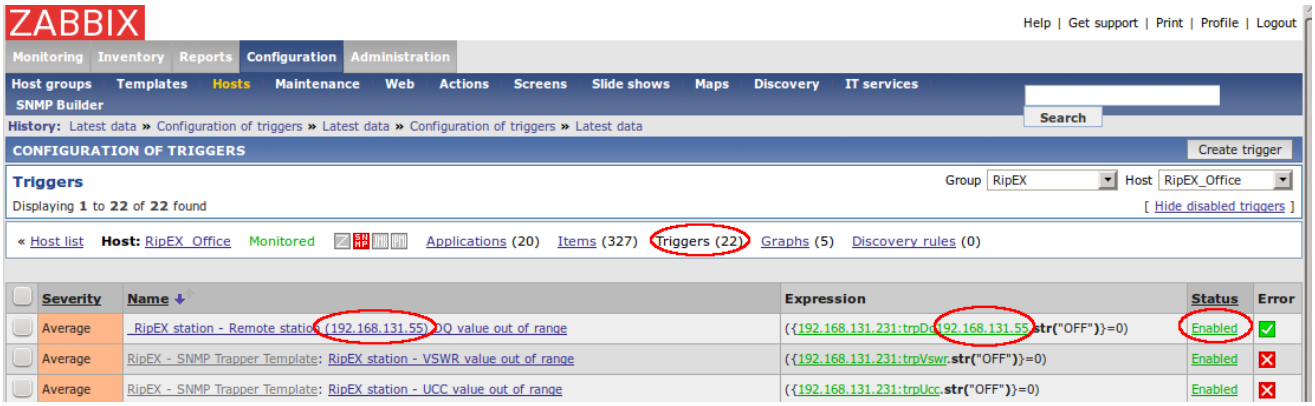

ZABBIX
Help | Get support | Print | Profile | Logout
Monitoring Inventory Reports Configuration Administration
Host groups Templates Hosts Maintenance Web Actions Screens Slide shows Maps Discovery IT services
SNMP Builder
Search
History: Latest data » Configuration of triggers » Latest data » Configuration of triggers » Latest data
CONFIGURATION OF TRIGGERS
Create trigger
Triggers
Group RipEX Host RipEX_Office
Displaying 1 to 22 of 22 found
[Hide disabled triggers]
« Host list Host: RipEX_Office Monitored Applications (20) Items (327) Triggers (22) Graphs (5) Discovery rules (0)
Severity Name Expression Status Error
Average RipEX station - Remote station (192.168.131.55) DQ value out of range ({192.168.131.231:trpDQ.192.168.131.55.str("OFF")}=0) Enabled
Average RipEX - SNMP Trapper Template: RipEX station - VSWR value out of range ({192.168.131.231:trpVswr.str("OFF")}=0) Enabled
Average RipEX - SNMP Trapper Template: RipEX station - UCC value out of range ({192.168.131.231:trpUcc.str("OFF")}=0) Enabled

Fig. 2.14: Enabled DQ trap trigger

Follow the same procedure (DQ and RSS) for other remote RipEX units as needed.

You can define Zabbix to send you an e-mail whenever any trap is triggered. See the Zabbix Documentation for e-mail configuration.

Now we just link the snmptrapd to use our script.

Please, find the file snmptrapd.conf, usually it's in the /etc/snmp/ directory. Edit or create the file as root with the following lines:

authCommunity execute public
authCommunity execute PUBLIC
traphandle default /bin/bash /home/zabbix/bin/snmptrap.sh

The first two lines will allow all received traps with community public or PUBLIC to be parsed and the third line will force the snmptrapd to use our script.

If you don't know what community names you will receive, add the following line to accept all community names.

disableAuthorization yes

Don't forget to restart the snmptrapd. Use -n to force snmptrapd not to translate SNMP OID numbers. Otherwise the script will NOT function correctly.

 # snmptrapd -n

Note

RipEX default Community string name is “public”, however it can be changed (since firmware release 1.3). All RipEX stations within the network must have the same Community string. Otherwise disableAuthorization has to be set to “yes” (or set authCommunity variables for all allowed Community string names).

Note

On Ubuntu releases, you can define default snmptrapd parameters at:

- */etc/default/snmpd:*
  - TRAPDRUN=yes

    - TRAPDOPTS='-n -p /var/run/snmptrapd.pid'
- */etc/init.d/snmpd:*
    - TRAPDRUN=yes
    - TRAPDOPTS='-n -p /var/run/snmptrapd.pid'

Basic Trap Functionality Tests

Now Zabbix is ready to receive SNMP traps from all RipEX stations and enter them into the database properly. In order to test it, force the trap to be sent from any RipEX and see whether it appears in the Zabbix front-end. If not, check that the respective UDP port (162) is enabled at your firewall and check the settings again. On the RipEX side you can monitor the interfaces to check whether the trap was sent (Monitoring) or by executing Tcpdump or Wireshark at the selected interface of your Zabbix server or somewhere along the intended packet path.

Another basic test can be run using the following command:

```
# zabbix_sender -z localhost -p 10051 -s "192.168.10.1" -k trpTemp -o "ALARM ON"
```

The IP address of your RipEX station is 192.168.10.1, key is “trpTemp” and the message for the Zabbix server is “TEST”. The command should trigger the host's “Temperature alarm”. Note that you need to have a host configured with this IP address, otherwise the trap will not be shown.

It is important to set the KEY value correctly, otherwise the trap would not match the trigger. See more KEY values with their description below:

- *trpRssIPAddress* - Remote station RSS value out of range (Replace the IPAddress with a real remote RipEX IP address)
- *trpDqIPAddress* - Remote station DQ value out of range (Replace the IPAddress with a real remote RipEX IP address)
- t*rpTxLost* - TX Lost value out of range
- *trpUcc* - UCC value out of range
- *trpTemp* - Modem temperature value out of range
- *trpRfpwr* - RF power value out of range
- *trpLanPr* - Ethernet RX/TX packet ratio out of range
- *trpCom1Pr* - COM1 RX/TX packet ratio out of range
- *trpCom2Pr* - COM2 RX/TX packet ratio out of range
- *trpHwIn* - HW input in the alarm state
- *trpHotStby* - Modem becomes active in a Hot-Standby mode
- *trpBpath* - Backup path state has changed
- *trpBpathAlt* - Alternative path state has changed
- *trpUnitReady* - Unit ready signal has changed

If you want to clear the trap alarm, just repeat the same zabbix_sender command, but change the message to contain the word "OFF". E.g. "ALARM OFF".

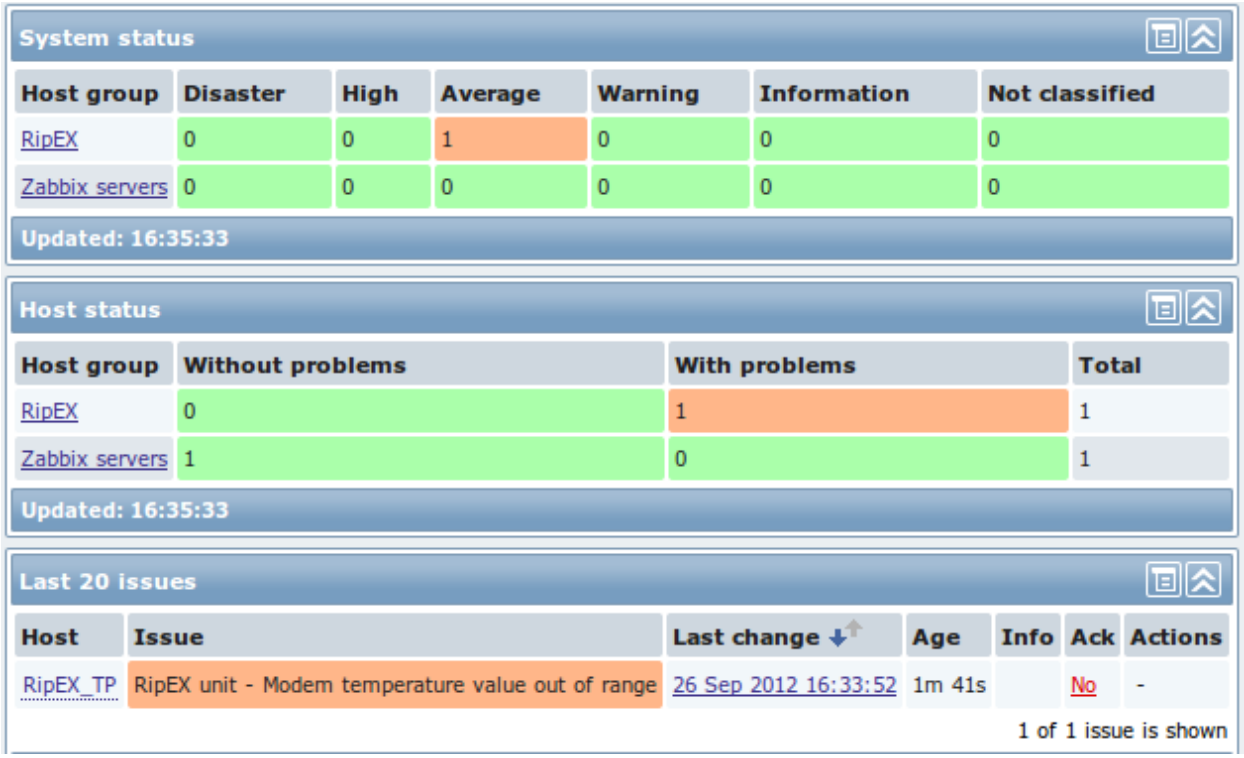


Fig. 2.15: Zabbix dashboard – Status of units

You can also see Trap's output in **Monitoring** → **Latest Data** → **TRAPS of your RipEX station** → **History**. The displayed information differs based on the trap received. See the detailed description in the respective Zabbix item.

2.5. What Else does Zabbix Offer?

There are many features provided by the Zabbix software. They are described in the Zabbix Documentation. Below are just a few of them.

You can create Screens. A Screen is a set of various graphs on one page for better overview of your network (temperature, UCC, RF power, …).

You can create Maps. If you administer many stations in many locations, a Map can be a good choice. You can define the background picture (e.g. real maps), various station pictures, station status, various statistics, etc. You can import any icon or background picture you want to use.

Fig. 2.16: Basic map with two RipEX stations

A short example of RipEX station configuration in Maps:

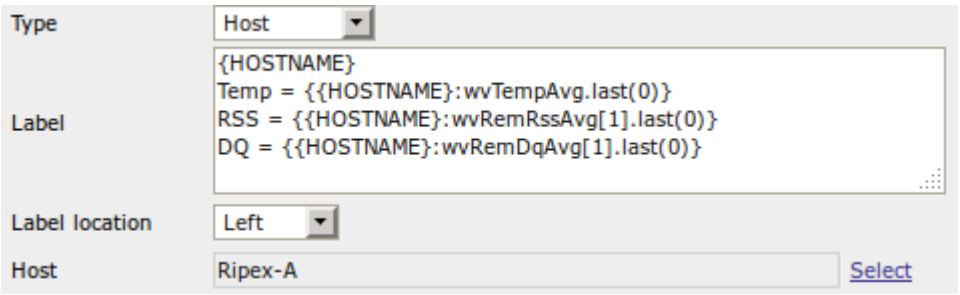


Fig. 2.17: Definition of RipEX station in maps

Each map can be divided into several sub-maps. It can be useful for various levels of detail.

You can configure **E-mail notifications** if anything goes wrong with the monitored stations. Just go to **Administration** → **Media Types** and define your SMTP server settings for outgoing messages. After that, go to **Administration** → **Users** to define who will be notified.

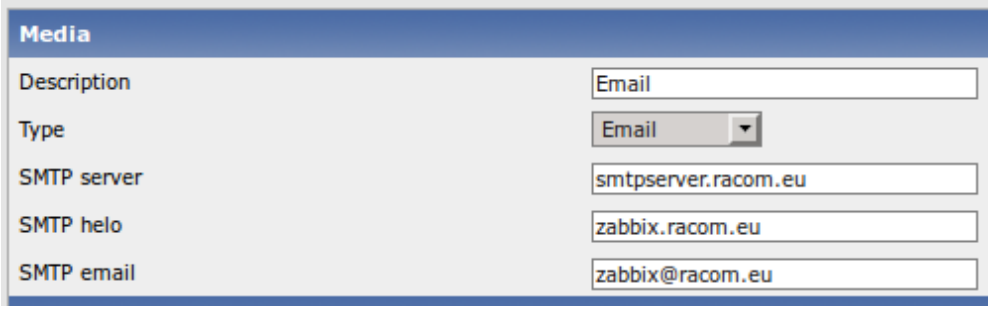


Fig. 2.18: E-mail configuration example

2.5.1. SNMP Builder

SNMP Builder is an add-on for Zabbix to help users to use SNMP OIDs as items or to browse the entire MIB tree hierarchy.

Note

The SNMP Builder is not supported in Zabbix 2.2.x releases.

Installation

Download the SNMP builder (.tar.gz)

- Zabbix version 2.0.x: https://github.com/atimonin/snmpbuilder

Within this site there is an installation documentation. Follow the steps described.

Simple guide for version 2.0.x (comments start with a double slash sign):

```
// $zabbix_frontend - path to Zabbix frontend (in Ubuntu /var/www/zabbix/)
// $your_src - directory with SNMP builder .tar.gz file
// $distname - name of the SNMP builder ditribution
1. $ cd $zabbix_frontend
2. $ tar xvzf $your_src/$distname/snmpbuilder-2.0_imgs.tar.gz
```

```
3. $ patch -p1 < $your_src/$distname/snmpbuilder-2.0.patch
4. $ cp -r $your_src/$distname/zabbix/* .
5. // Restart the Apache service:
          # service apache2 restart  OR
          # /etc/init.d/apache2 restart
```

If you have been successful, the SNMP Builder folder is added in the Zabbix front-end:

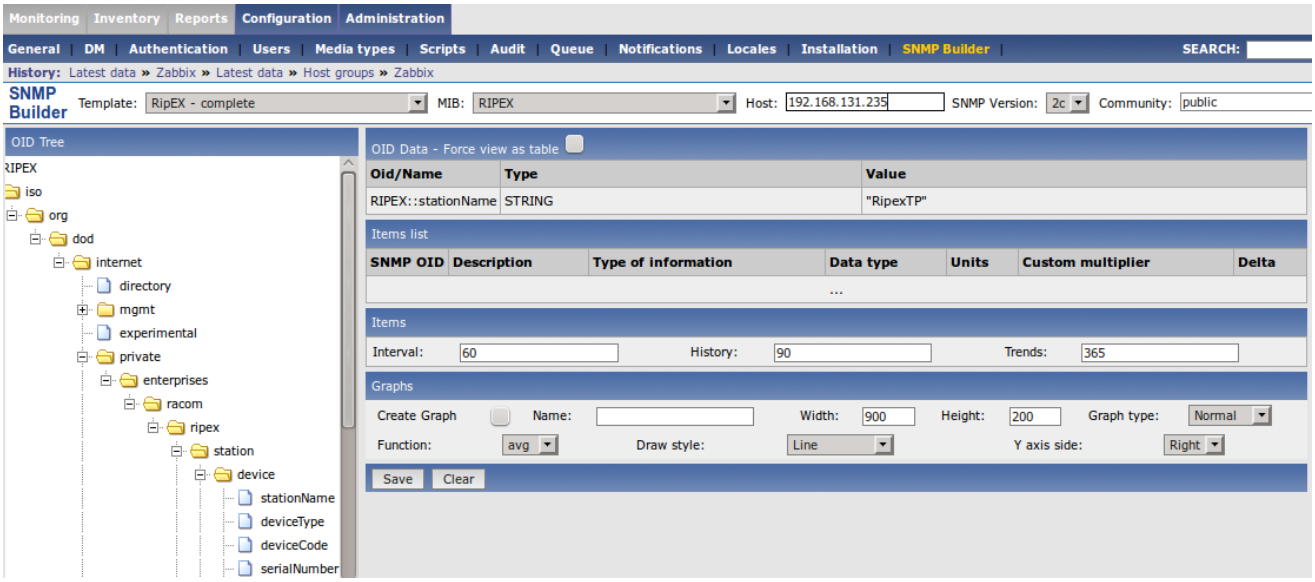


Fig. 2.19: SNMP Builder overview

Note

The step three (patch) can end with an error message - in such a case, edit the `jsLoader.php` file manually. Follow the instructions in the `jsLoader.php.rej` file (probably adding lines with a plus sign on the beginning).

How to Work with the SNMP Builder and the RipEX MIB?

To be able to see the RipEX MIB table, you need to import it to the Zabbix server default MIB directory – usually /usr/share/snmp/mibs/. The directory is shown at the SNMP Builder front-end in the second menu “MIB”.

Front-end settings:

- *Template:* choose the RipEX template
- *MIB:* choose the RipEX MIB. If you cannot see any MIB, you probably have a wrong path to the MIB directory – change variables `$mibs_dir` and `MIBS_ALL_PATH` accordingly in the `$zabbix_frontend/snmp_builder.php` file.
- *Host:* Fill in the RipEX IP address
- *SNMP Version:* 2c
- *Community:* public

On the left side of the Zabbix front-end, you can see the OID Tree. You can navigate in there as you wish. If you click on any OID object, it should print a returned value on the right. By selecting these values, you can easily create various items with basic settings. Not all options are implemented in the SNMP Builder.

If you have imported the RipEX template, you have all OID objects already transformed into the Zabbix items.

Note

RipEX supports several OID objects from the RS232 MIB and the RFC1213 MIB tables. These are contained in the http://www.racom.eu/download/hw/ripex/free/eng/3_fw/RACOM-RipEX-MIB.zip file.

2.6. RipEX MIB Table

2.6.1. RipEX

OID	Name	Access	State	Description
33555.2.1.1.1	stationName	read-only	current	Station Name.
33555.2.1.1.2	deviceType	read-only	current	Device type.
33555.2.1.1.3	deviceCode	read-only	current	Device code.
33555.2.1.1.4	serialNumber	read-only	current	Product serial number.
33555.2.1.1.5	deviceMode	read-only	current	Station working mode.
33555.2.1.1.6.1	hwVerModem	read-only	current	Modem HW version.
33555.2.1.1.6.2	hwVerRadio	read-only	current	Radio HW version.
33555.2.1.1.7.1	swVermodem	read-only	current	Modem firmware version.
33555.2.1.1.7.2	swVerSDDR	read-only	current	SDDR firmware version.
33555.2.1.1.7.3	swVerDriver	read-only	current	Driver firmware version.
33555.2.1.1.7.4	swVerBootloader	read-only	current	Bootloader version.
33555.2.2.1.1	rRxFrequency	read-only	current	Radio interface Rx frequency in Hz.
33555.2.2.1.2	rTxFrequency	read-only	current	Radio interface Tx frequency in Hz.
33555.2.2.1.3	rRfPwr	read-only	current	Radio interface RF Power.
33555.2.2.1.4	rEncryption	read-only	current	Radio interface encryption method.
33555.2.2.1.5	rFEC	read-only	current	Radio interface FEC state.
33555.2.2.2.1	eGateway	read-only	current	Ethernet interface gateway address.
33555.2.2.2.2	eDhcp	read-only	current	Ethernet interface DHCP mode.
33555.2.2.2.3	eShaping	read-only	current	Ethernet interface shaping state.
33555.2.2.2.4	eBCastMCast	read-only	current	Ethernet interface broadcast and multicast state.
33555.2.2.2.5	eSpeed	read-only	current	Ethernet interface bit rate and duplex settings.
33555.2.2.3.1	ifTmEnable	read-only	current	TCP Modbus state.
33555.2.2.3.2	ifTmPort	read-only	current	TCP Modbus port.

33555.2.2.3.3	ifTmTimeout	read-only	current	TCP Modbus socket timeout in seconds.
33555.2.2.3.4	ifTmBCast	read-only	current	TCP Modbus COM protocol broadcast accept.
33555.2.2.3.5	ifTmATM	read-only	current	TCP Modbus COM protocol address translation mode.
33555.2.2.4.1	ifTsEnable	read-only	current	Terminal server state.
33555.2.2.4.2	ifTsNumber	read-only	current	Number of Terminal server interfaces.
33555.2.2.4.3	ifTsTable	not-accessible	current	List of Terminal server interface entries.
33555.2.2.4.3.1	ifTsEntry	not-accessible	current	Terminal server interface entry.
33555.2.2.4.3.1.1.X	tsIndex	read-only	current	Unique index for each interface.
33555.2.2.4.3.1.2.X	tsEnable	read-only	current	Terminal server interface state.
33555.2.2.4.3.1.3.X	tsEthProtType	read-only	current	Terminal server Ethernet protocol type.
33555.2.2.4.3.1.4.X	tsEthProtTimeout	read-only	current	Terminal server Ethernet protocol socket timeout in seconds.
33555.2.2.4.3.1.5.X	tsEthProtMyPort	read-only	current	Terminal server Ethernet protocol socket TCP/UDP port.
33555.2.2.4.3.1.6.X	tsEthProtDestIP	read-only	current	Terminal server partner's IP address.
33555.2.2.4.3.1.7.X	tsEthProtDestPort	read-only	current	Terminal server partner's destination TCP/UDP port.
33555.2.2.4.3.1.8.X	tsComProtType	read-only	current	Terminal server COM user protocol type.
33555.2.2.5.1	ifComNumber	read-only	current	Number of COM interfaces.
33555.2.2.5.2	ifComTable	not-accessible	current	List of COM interface entries.
33555.2.2.5.2.1	ifComEntry	not-accessible	current	COM interface entry.
33555.2.2.5.2.1.1.X	comIndex	read-only	current	Unique index for each interface.
33555.2.2.5.2.1.2.X	comIdle	read-only	current	COM interface idle in bytes.
33555.2.2.5.2.1.3.X	comMtu	read-only	current	COM interface MTU in bytes.
33555.2.2.5.2.1.4.X	comProtocol	read-only	current	COM interface protocol.
33555.2.2.21.1	ifHwAInputType	read-only	current	HW alarm input contact type.
33555.2.2.21.2	ifHwAInputState	read-only	current	HW alarm input contact state.
33555.2.3.1.1.1	stRadioTotDuplicates	read-only	current	Total radio duplicate packets counter.
33555.2.3.1.1.2	stRadioTotRepeats	read-only	current	Total radio repeated packets counter.
33555.2.3.1.1.3	stRadioTotLost	read-only	current	Total radio lost packets counter.
33555.2.3.1.1.4	stRadioTotCtlPacketsRx	read-only	current	Total Rx radio control packets counter.
33555.2.3.1.1.5	stRadioTotCtlPacketsTx	read-only	current	Total Tx radio control packets counter.

33555.2.3.1.1.6	stRadioTot-DataErr	read-only	current	Total radio data error packets counter.
33555.2.3.1.1.7	stRadioTotRejec-ted	read-only	current	Total radio rejected packets counter.
33555.2.3.1.1.8	stRadioTotPacket-sRx	read-only	current	Remote station total Rx packets counter.
33555.2.3.1.1.9	stRadioTotPacket-sTx	read-only	current	Remote station total Tx packets counter.
33555.2.3.1.1.10	stRadioTotBytes-Rx	read-only	current	Remote station total Rx bytes counter.
33555.2.3.1.1.11	stRadioTot-BytesTx	read-only	current	Remote station total Tx bytes counter.
33555.2.3.1.1.12	stRadioTotIpErr	read-only	current	Total radio IP error packets counter.
33555.2.3.1.1.13	stRadioTotSub-HeadErr	read-only	current	Total radio subheader error packets counter.
33555.2.3.1.1.14	stRadioTotHead-Err	read-only	current	Total radio header error packets counter.
33555.2.3.1.1.15	stRadioTotFalse-Sync	read-only	current	Total radio false sync counter.
33555.2.3.1.2	stRadioRemNum-ber	read-only	current	Number of remote stations.
33555.2.3.1.3	stRadioRemTable	not-access-ible	current	List of remote station entries.
33555.2.3.1.3.1	stRadioRemEntry	not-access-ible	current	Radio remote station entry.
33555.2.3.1.3.1.1.X	stRemIndex	read-only	current	Remote station index.
33555.2.3.1.3.1.2.X	stRemIpAddr	read-only	current	Remote station IP address.
33555.2.3.1.3.1.3.X	stRemPacketsRx	read-only	current	Remote station Rx packets counter.
33555.2.3.1.3.1.4.X	stRemPacketsTx	read-only	current	Remote station Tx packets counter.
33555.2.3.1.3.1.5.X	stRemBytesRx	read-only	current	Remote station Rx bytes counter.
33555.2.3.1.3.1.6.X	stRemBytesTx	read-only	current	Remote station Tx bytes counter.
33555.2.3.1.3.1.7.X	stRemDuplicates	read-only	current	Remote station duplicate packets counter.
33555.2.3.1.3.1.8.X	stRemRepeats	read-only	current	Remote station repeated packets counter.
33555.2.3.1.3.1.9.X	stRemLost	read-only	current	Remote station lost packets counter.
33555.2.3.1.3.1.10.X	stRemCtlPackets-Rx	read-only	current	Remote station Rx radio control packets counter.
33555.2.3.1.3.1.11.X	stRemCtlPacket-sTx	read-only	current	Remote station Tx radio control packets counter.
33555.2.3.1.3.1.12.X	stRemDataErr	read-only	current	Remote station data error packets counter.
33555.2.3.1.3.1.13.X	stRemRejected	read-only	current	Remote station rejected packets counter.

33555.2.3.1.3.1.14.X	stRemTotalPacket-sRx	read-only	current	Remote station total Rx packets counter.
33555.2.3.1.3.1.15.X	stRemTotalPacket-sTx	read-only	current	Remote station total Tx packets counter.
33555.2.3.1.3.1.16.X	stRemTotalBytes-Rx	read-only	current	Remote station total Rx bytes counter.
33555.2.3.1.3.1.17.X	s t R e m T o t a l - BytesTx	read-only	current	Remote station total Tx bytes counter.
33555.2.3.2.1	stTcpModNumber	read-only	current	Number of TCP Modbus ports.
33555.2.3.2.2	stTcpModTable	not-access-ible	current	List of TCP Modbus port entries.
33555.2.3.2.2.1	stTcpModEntry	not-access-ible	current	TCP Modbus port entry.
33555.2.3.2.2.1.1.X	stTcpModIndex	read-only	current	TCP Modbus port index.
33555.2.3.2.2.1.2.X	stTcpModPackets-Rx	read-only	current	TCP Modbus Rx packets counter.
33555.2.3.2.2.1.3.X	stTcpModPacket-sTx	read-only	current	TCP Modbus Tx packets counter.
33555.2.3.2.2.1.4.X	stTcpModBytesRx	read-only	current	TCP Modbus Rx bytes counter.
33555.2.3.2.2.1.5.X	stTcpModBytesTx	read-only	current	TCP Modbus Tx bytes counter.
33555.2.3.3.1	stTermServNum-ber	read-only	current	Number of Terminal Server ports.
33555.2.3.3.2	stTermServTable	not-access-ible	current	List of Terminal Server port entries.
33555.2.3.3.2.1	stTermServEntry	not-access-ible	current	Terminal Server port entry.
33555.2.3.3.2.1.1.X	stTermServIndex	read-only	current	Terminal Server port index.
33555.2.3.3.2.1.2.X	stTermServPack-etsRx	read-only	current	Terminal Server Rx packets counter.
33555.2.3.3.2.1.3.X	stTermServPack-etsTx	read-only	current	Terminal Server Tx packets counter.
33555.2.3.3.2.1.4.X	stTermServBytes-Rx	read-only	current	Terminal Server Rx bytes counter.
33555.2.3.3.2.1.5.X	s t T e r m S e r v - BytesTx	read-only	current	Terminal Server Tx bytes counter.
33555.2.3.4.1	stComNumber	read-only	current	Number of COM ports.
33555.2.3.4.2	stComTable	not-access-ible	current	List of COM port entries.
33555.2.3.4.2.1	stComEntry	not-access-ible	current	COM port entry.
33555.2.3.4.2.1.1.X	stComIndex	read-only	current	COM port index.
33555.2.3.4.2.1.2.X	stComPacketsRx	read-only	current	COM Rx packets counter.
33555.2.3.4.2.1.3.X	stComPacketsTx	read-only	current	COM Tx packets counter.
33555.2.3.4.2.1.4.X	stComBytesRx	read-only	current	COM Rx bytes counter.

33555.2.3.4.2.1.5.X	stComBytesTx	read-only	current	COM Tx bytes counter.
33555.2.3.5.1	stTcpProxyNumber	read-only	current	Number of TCP proxy ports.
33555.2.3.5.2	stTcpProxyTable	not-accessible	current	List of TCP proxy port entries.
33555.2.3.5.2.1	stTcpProxyEntry	not-accessible	current	TCP proxy port entry.
33555.2.3.5.2.1.1.X	stTcpProxyIndex	read-only	current	TCP proxy port index.
33555.2.3.5.2.1.2.X	stTcpProxyPacketsRx	read-only	current	TCP proxy Rx packets counter.
33555.2.3.5.2.1.3.X	stTcpProxyPacketsTx	read-only	current	TCP proxy Tx packets counter.
33555.2.3.5.2.1.4.X	stTcpProxyBytesRx	read-only	current	TCP proxy Rx bytes counter.
33555.2.3.5.2.1.5.X	stTcpProxyBytesTx	read-only	current	TCP proxy Tx bytes counter.
33555.2.4.1.5	wvTxLostLast	read-only	current	Local station - Last Tx lost value in %.
33555.2.4.1.6	wvTxLostAvg	read-only	current	Local station - Average Tx lost value in hundredths of %.
33555.2.4.1.7	wvUccLast	read-only	current	Local station - Last UCC value in tenths of Volt (V).
33555.2.4.1.8	wvUccAvg	read-only	current	Local station - Average UCC value in thousandths of Volt (V).
33555.2.4.1.9	wvTempLast	read-only	current	Local station - Last device temperature value in tenths of Celsius (C).
33555.2.4.1.10	wvTempAvg	read-only	current	Local station - Average device temperature value in thousandths of Celsius (C).
33555.2.4.1.11	wvRfPwrLast	read-only	current	Local station - Last RF power value in tenths of Watt (W).
33555.2.4.1.12	wvRfPwrAvg	read-only	current	Local station - Average RF power value in thousandths of Watt (W).
33555.2.4.1.13	wvVswrLast	read-only	current	Local station - Last VSWR value from interval <3, 25> in tenths.
33555.2.4.1.14	wvVswrAvg	read-only	current	Local station - Average VSWR value from interval <300, 2500> in thousandths.
33555.2.4.1.41	wvRxTxEth	read-only	current	Local station - ETH interface Rx to Tx packets ratio value from interval <1, 10000> in hundredths.
33555.2.4.1.42	wvRxTxCom1	read-only	current	Local station - COM1 interface Rx to Tx packets ratio value from interval <1, 10000> in hundredths.

33555.2.4.1.43	wvRxTxCom2	read-only	current	Local station - COM2 interface Rx to Tx packets ratio value from interval <1, 10000> in hundredths.
33555.2.4.2	wvRemoteNumber	read-only	current	Number of remote stations.
33555.2.4.3	wvRemoteTable	not-accessible	current	List of remote stations.
33555.2.4.3.1	wvRemoteEntry	not-accessible	current	Remote station watched values entry.
33555.2.4.3.1.1.X	wvRemIndex	read-only	current	Remote station - Unique index.
33555.2.4.3.1.2.X	wvRemIpAddr	read-only	current	Remote station - IP address.
33555.2.4.3.1.3.X	wvRemHearings	read-only	current	Remote station - Total heard packets from the remote station.
33555.2.4.3.1.4.X	wvRemRssLast	read-only	current	Remote station - Last RSS value in dBm.
33555.2.4.3.1.5.X	wvRemRssAvg	read-only	current	Remote station - Average RSS value in hundredths of dBm.
33555.2.4.3.1.6.X	wvRemDqLast	read-only	current	Remote station - Last DQ value.
33555.2.4.3.1.7.X	wvRemDqAvg	read-only	current	Remote station - Average DQ value in hundredths.
33555.2.4.3.1.12.X	wvRemTxLostLast	read-only	current	Remote station - Last Tx lost value in %.
33555.2.4.3.1.13.X	wvRemTxLostAvg	read-only	current	Remote station - Average Tx lost value in hundredths of %.
33555.2.4.3.1.14.X	wvRemUccLast	read-only	current	Remote station - Last UCC value in tenths of Volt (V).
33555.2.4.3.1.15.X	wvRemUccAvg	read-only	current	Remote station - Average UCC value in thousandths of Volt (V).
33555.2.4.3.1.16.X	wvRemTempLast	read-only	current	Remote station - Last device temperature value in tenths of Celsius (C).
33555.2.4.3.1.17.X	wvRemTempAvg	read-only	current	Remote station - Average device temperature value in thousandths of Celsius (C).
33555.2.4.3.1.18.X	wvRemRfPwrLast	read-only	current	Remote station - Last RF power value in tenths of Watt (W).
33555.2.4.3.1.19.X	wvRemRfPwrAvg	read-only	current	Remote station - Average RF power value in thousandths of Watt (W).
33555.2.4.3.1.20.X	wvRemVswrLast	read-only	current	Remote station - Last VSWR value from interval <3, 25> in tenths.
33555.2.4.3.1.21.X	wvRemVswrAvg	read-only	current	Remote station - Average VSWR value from interval <300, 2500> in thousandths.
33555.2.5.1.1	alarmThrRssMin	read-only	current	Alarm threshold - minimum - RSS value in dBm.

33555.2.5.1.2	alarmThrRssMax	read-only	current	Alarm threshold - maximum - RSS value in dBm.
33555.2.5.1.3	alarmThrDqMin	read-only	current	Alarm threshold - minimum - DQ value.
33555.2.5.1.4	alarmThrDqMax	read-only	current	Alarm threshold - maximum - DQ value.
33555.2.5.1.9	alarmThrTxLost-Min	read-only	current	Alarm threshold - minimum - Tx lost value in %.
33555.2.5.1.10	alarmThrTxLost-Max	read-only	current	Alarm threshold - maximum - Tx lost value in %.
33555.2.5.1.11	alarmThrUccMin	read-only	current	Alarm threshold - minimum - UCC value in tenths of Volt (V).
33555.2.5.1.12	alarmThrUccMax	read-only	current	Alarm threshold - maximum - UCC value in tenths of Volt (V).
33555.2.5.1.13	alarmThrTempMin	read-only	current	Alarm threshold - minimum - device temperature value in tenths of Celsius (C).
33555.2.5.1.14	alarmThrTemp-Max	read-only	current	Alarm threshold - maximum - device temperature value in tenths of Celsius (C).
33555.2.5.1.15	alarmThrRfP-wrMin	read-only	current	Alarm threshold - minimum - RF power value in tenths of Watt (W).
33555.2.5.1.16	alarmThrRfP-wrMax	read-only	current	Alarm threshold - maximum - RF power value in tenths of Watt (W).
33555.2.5.1.17	alarmThrVswrMin	read-only	current	Alarm threshold - minimum - VSWR value from interval <3, 25> in tenths.
33555.2.5.1.18	alarmThrVswrMax	read-only	current	Alarm threshold - maximum - VSWR value from interval <3, 25> in tenths.
33555.2.5.1.31	alarmThrRx-TxEthMin	read-only	current	Alarm threshold - minimum - ETH interface Rx to Tx packets ratio value in hundredths.
33555.2.5.1.32	alarmThrRxTxEth-Max	read-only	current	Alarm threshold - maximum - ETH interface Rx to Tx packets ratio value in hundredths.
33555.2.5.1.33	alarmThrRxTx-Com1Min	read-only	current	Alarm threshold - minimum - COM1 interface Rx to Tx packets ratio value in hundredths.
33555.2.5.1.34	alarmThrRxTx-Com1Max	read-only	current	Alarm threshold - maximum - COM1 interface Rx to Tx packets ratio value in hundredths.
33555.2.5.1.35	alarmThrRxTx-Com2Min	read-only	current	Alarm threshold - minimum - COM2 interface Rx to Tx packets ratio value in hundredths.
33555.2.5.1.36	alarmThrRxTx-Com2Max	read-only	current	Alarm threshold - maximum - COM2 interface Rx to Tx packets ratio value in hundredths.
33555.2.5.2.1	alarmStateRss	read-only	current	Alarm state - RSS.
33555.2.5.2.2	alarmStateDq	read-only	current	Alarm state - DQ.

33555.2.5.2.5	alarmStateTxLost	read-only	current	Alarm state - Tx lost.
33555.2.5.2.6	alarmStateUcc	read-only	current	Alarm state - UCC.
33555.2.5.2.7	alarmStateTemp	read-only	current	Alarm state - device temperature.
33555.2.5.2.8	alarmStateRfPwr	read-only	current	Alarm state - RF power.
33555.2.5.2.9	alarmStateVswr	read-only	current	Alarm state - VSWR.
33555.2.5.2.16	alarmStateRx-TxEth	read-only	current	Alarm state - ETH interface Rx to Tx packets ratio.
33555.2.5.2.17	alarmStateRxTx-Com1	read-only	current	Alarm state - COM1 interface Rx to Tx packets ratio.
33555.2.5.2.18	alarmStateRxTx-Com2	read-only	current	Alarm state - COM2 interface Rx to Tx packets ratio.
33555.2.5.2.31	alarmStateHwIn-put	read-only	current	Alarm state - HW Input.
33555.2.5.2.32	alarm-StateUnitReady	read-only	current	Alarm state - Unit ready.
33555.2.6.1	bpathsNumber	read-only	current	Number of Backup Paths.
33555.2.6.2	bpathsTable	not-access-ible	current	List of Backup Paths entries.
33555.2.6.2.1	bpathsEntry	not-access-ible	current	Backup Paths entry.
33555.2.6.2.1.1.X	bpathsIndex	read-only	current	Backup Paths index.
33555.2.6.2.1.2.X	bpathsPeerIp	read-only	current	Backup Paths - Peer IP address.
33555.2.6.2.1.3.X	bpathsName	read-only	current	Backup Paths - Symbolic Name.
33555.2.6.2.1.4.X	bpath-sAltUsedPrio	read-only	current	Backup Paths - Alternative Paths - Currently used path priority.
33555.2.6.2.1.5.X	bpathsAltUsedGw	read-only	current	Backup Paths - Alternative Paths - Currently used path Gateway IP address.
33555.2.6.2.1.6.X	bpathsAltUsed-State	read-only	current	Backup Paths - Alternative Paths - Currently used path State.
33555.2.6.2.1.7.X	bpathsAltPass-iveState	read-only	current	Backup Paths - Alternative Paths - Currently passive paths State.
33555.2.10.1	trpRss		current	A notification to indicate that average RSS value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateRss: RSS alarm state. - wvRemRssAvg: Remote station - Average RSS value in hundredths of dBm. - wvRemIpAddr: Remote station IP address.
33555.2.10.2	trpDq		current	A notification to indicate that average DQ value has exceeded threshold limits. This notification sends additional information about the event by including the

				following objects in its varbinding list. - alarmStateDq: DQ alarm state. - wvRemDqAvg: Remote station - Average DQ value in hundredths. - wvRemIpAddr: Remote station IP address.
33555.2.10.5	trpTxLost		current	A notification to indicate that average Tx lost value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateTxLost: Tx lost alarm state. - wvTxLostAvg: Local station - Average Tx lost value in hundredths of %.
33555.2.10.6	trpUcc		current	A notification to indicate that average UCC value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateUcc: UCC alarm state. - wvUccAvg: Local station - Average UCC value in thousandths of Volt (V).
33555.2.10.7	trpTemp		current	A notification to indicate that average device temperature value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateTemp: Device temperature alarm state. - wvTempAvg: Local station - Average device temperature value in thousandths of Celsius (C).
33555.2.10.8	trpRfPwr		current	A notification to indicate that average RF power value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateRfPwr: RF power alarm state. - wvRfPwrAvg: Local station - Average RF power value in thousandths of Watt (W).
33555.2.10.9	trpVswr		current	A notification to indicate that average VSWR value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateVswr: VSWR alarm state. - wvVswrAvg: Local station - Aver-

				age VSWR value from interval <300, 2500> in thousandths.
33555.2.10.10	trpEthPr		current	A notification to indicate that average ETH interface Rx to Tx packets ratio value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateRxTxEth: Alarm state of ETH Rx to Tx packets ratio value. - wvRxTxEth: Local station - ETH Rx to Tx packets ratio value from interval <1, 10000> in hundredths.
33555.2.10.11	trpCom1Pr		current	A notification to indicate that average COM1 interface Rx to Tx packets ratio value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateRxTxCom1: Alarm state of COM1 Rx to Tx packets ratio value. - wvRxTxCom1: Local station - COM1 Rx to Tx packets ratio value from interval <1, 10000> in hundredths.
33555.2.10.12	trpCom2Pr		current	A notification to indicate that average COM2 interface Rx to Tx packets ratio value has exceeded threshold limits. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateRxTxCom2: Alarm state of COM2 Rx to Tx packets ratio value. - wvRxTxCom2: Local station - COM2 Rx to Tx packets ratio value from interval <1, 10000> in hundredths.
33555.2.10.13	trpHwIn		current	A notification to indicate that HW alarm input state has changed. This notification sends additional information about the event by including the following objects in its varbinding list. - ifHwAInputState: HW alarm input contact state. - ifHwAInputType: HW alarm input contact type.
33555.2.10.14	trpHotStby		current	A notification to indicate that device in Hot Standby mode has been activated. This notification sends additional information about the event by including the following objects in its varbinding list. - serialNumber: Product serial number. - stationName: Station name.

33555.2.10.15	trpBpath		current	A notification to indicate a change in Backup paths system - backup path state has changed. This notification sends additional information about the event by including the following objects in its varbinding list. - bpathsPeerIp: Backup path peer IP address. - bpathsName: Backup path symbolic name. - bpathsAltUsedPrio: Currently used alternative path priority number. - bpathsAltUsedGw: Currently used alternative path gateway IP address. - bpathsAltUsedState: Currently used alternative path state.
33555.2.10.16	trpBpathAlt		current	A notification to indicate a change in Backup paths system - alternative path state has changed. This notification sends additional information about the event by including the following objects in its varbinding list. - bpathsPeerIp: Backup path peer IP address. - bpathsName: Backup path symbolic name. - bpathsAltUsedPrio: Currently used alternative path priority number. - bpathsAltUsedGw: Currently used alternative path gateway IP address. - bpathsAltUsedState: Currently used alternative path state.
33555.2.10.17	trpUnitReady		current	A notification to indicate that Unit ready signal has changed. This notification sends additional information about the event by including the following objects in its varbinding list. - alarmStateUnitReady: alarm input state.

3. Data speed and Modulations

On efficient use of narrowband radio channel

Introduction

The industrial narrowband *land mobile radio* (LMR) devices, as considered in this paper, have been the subject to European standard ETSI EN 300 113 [1]. The system operates on frequencies between 30 MHz and 1 GHz, with channel separations of up to 25 kHz, and is intended for private, fixed, or mobile, radio packet switching networks. Data telemetry, SCADA, maritime and police radio services; traffic monitoring; gas, water, and electricity producing factories are the typical system applications. Long distance coverage, high power efficiency, and efficient channel access techniques in half duplex operation are the primary advantages the system relays on. Very low level of adjacent channel power emissions and robust radio receiver architectures, with high dynamic range, enable for a system's co-existence with various communication standards without the additional guard band frequency intervals.

On the other hand, the strict limitations of the referenced standard as well as the state of the technology, has hindered the increase in spectrum efficiency, with which the system has used its occupied bandwidth. With its modification as well as with the new emerging specifications (ETSI EN 302 561 [2], ETSI EN 301 166 [3]) it is now possible for the up-to-date architectures of narrowband LMR devices to make the utilization of more efficient modes of system operation practically applicable.

The main objective of this paper is to describe the favorable properties of operational modes based on advanced nonlinear and linear digital modulation techniques in order to easy the decision on their usage and thus to help system integrators to increase the efficiency of the narrowband radio channel utilization allocated to the new generation of industrial LMR devices.

3.1. Narrowband radio transmitter

From the very advent of the radio transmission, it was evident that a radio device should not only use its occupied channel bandwidth effectively, but, in addition, should also avoid any unnecessary interference with other systems. Since then the frequency spectrum had been proving its importance and has become a scarce resource nowadays.

The narrowband radio devices under consideration are specified mostly by the European standard ETSI EN 300 113 [1]. Such radio equipments have to face challenging environmental and radio conditions all over the world. The dynamic range in the vicinity of 100 dB, very strict adjacent channel transmitted power attenuation requirements, high data sensitivity, adjacent channel selectivity, high level of radio blocking or desensitization and high co-channel rejection [1], are its most important radio characteristics to mention. It is no wonder that for such high dynamic range demands, super heterodyne transceiver architectures with a majority of analog components are still widely used. But yet the radio transceiver has to be small in dimensions, consumes low power and maintains all its parameters over the wide industrial temperature range and over extensive period of time for reasonable price. At the same time, it should provide enough flexibility to accommodate different channel bandwidths, digital modulation formats, data rates, and techniques, to combat negative effects of radio channel. From this point of view, the *software defined radio* (SDR) concept is, indisputably, a prospective alternative and has not been widely used by these systems. The rapid expansion of the digital signal processing, together with the advancements in signal analog-to-digital converters technology have, in recent years, made such projects economically feasible.

Today's LMR systems, being subject to [1], use mostly exponential constant envelope modulations GMSK, 2-CPFSK and 4-CPFSK. The application of the continuous phase modulations is mainly due

to the extreme *adjacent channel transmitted power* (ACP) attenuation requirements, and inherent robustness against channel nonlinearities. Relatively simple implementation of non-coherent demodulators and synchronization algorithms also significantly contributes to the efficient channel usage, especially in packet-based switching networks. The systems thus maintain good power efficiency while the spectral efficiency reaches compromising values not exceeding 1 bit/s/Hz.

3.1.1. Digital modulation for narrowband channel

The prime classification of the digital modulation techniques into a *nonlinear* (or *exponential*) and *linear* modulation class is based on the way how the modulated signal has been generated. The complex modulation envelope of the linearly modulated signal such as M-PSK, M-QAM etc. can be described by a linear superposition of the properly filtered modulation impulses weighted by the information symbols. In case of the nonlinear modulation techniques, this general rule is valid only for the modulation signal which modulates the phase of the fundamental carrier signal. Thus the modulation process itself is nonlinear, exponential. The M-CPFSK in this case is recognized as a general class of nonlinear or exponential digital modulation with a continuous phase change.

3.1.2. Adjacent channel power and spectrum efficiency

The adjacent channel power or *adjacent channel interference* (ACI) is that part of the total output power of a transmitter under defined conditions of modulation, which falls within a specified pass-band centred on the nominal frequency of either of the adjacent channels. This power is the sum of the mean power produced by the modulation, hum and noise of the transmitter. Adjacent channel power is usually referenced to the unmodulated carrier power [1]:

> *For a channel separation of 25 kHz, the adjacent channel power shall not exceed a value of **60 dB** below the transmitter power without the need to be below -37 dBm.*

It is interesting to note that, until 07/2007, the standard strictly demanded the adjacent channel power ratio of -70 dB.

The ACP parameter is particularly important in LMR systems, since it influences the density of the radio channels that can be used in a given area. Its value originated in the use of the traditional analog *frequency modulated* (FM) radio systems. Ironically, it was one of the main limitations for why those systems were – for many years – not able to utilize spectrally more efficient modulation schemes. The problem in this case is that all the advanced multi-level modulation techniques such as M-PSK, M-QAM, OFDM, CDMA or FBMCM have one negative property and that is a non-constant modulation envelope.

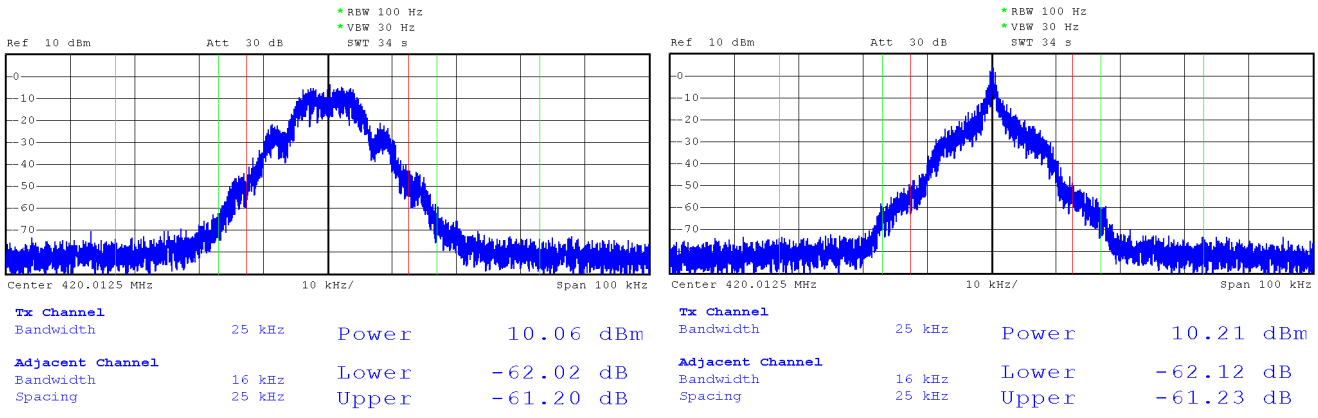


*Fig. 3.1: Modulated signal spectrums. (**left**) 2CPFSK with R=10.4 kBaud, modulation index h~0.6. (**right**) 2CPFSK with R=17.3 kBaud, modulation index h~0.2. 30 dB attenuator used in series.*

In the systems, where the transmitter power efficiency is of high importance, the *transmitter nonlinearity* also creates an important issue. Generally speaking, the higher the transmitter nonlinearity, the higher the transmitter efficiency can be reached. Unfortunately, the device with a nonlinear transfer function also tends to distort the spectrum of the transmitted signal, especially if the modulated signal exhibits the non-constant modulation envelope. In contrast, it is also true that only the non-constant envelope modulation can withstand a strict band limitation by means of modulation filtering – characterized by the roll-off parameter α in the following text. In other words, if the signal has a constant modulation envelope, it has an unlimited spectrum, and, if it has a band limited spectrum, it experiences the amplitude variations, which after passing through the nonlinear power amplifier, would be suppressed, but would also regenerate the side-lobes of the modulated signal spectrum. The phenomenon is known as the spectral *re-growth*, and it depends mainly on the three transmitter characteristics. Those are *peak to average power ratio* (PAPR) of the digital modulation scheme in use, *transmitter nonlinearity* and *the efficiency of the power amplifier linearization* or *pre-distortion technique* and all have to be considered when selecting the digital modulation technique for the system, where both power and spectrum are the key issues.

In light of these facts one can arrive at the conclusion that setting up the limit at −60 dB[1] rather than −70 dB was a reasonable step, while the initial limit has been left to be beyond the state of the present linearization technology for equipments production which in turn hindered the use of spectrally more efficient modulation techniques.

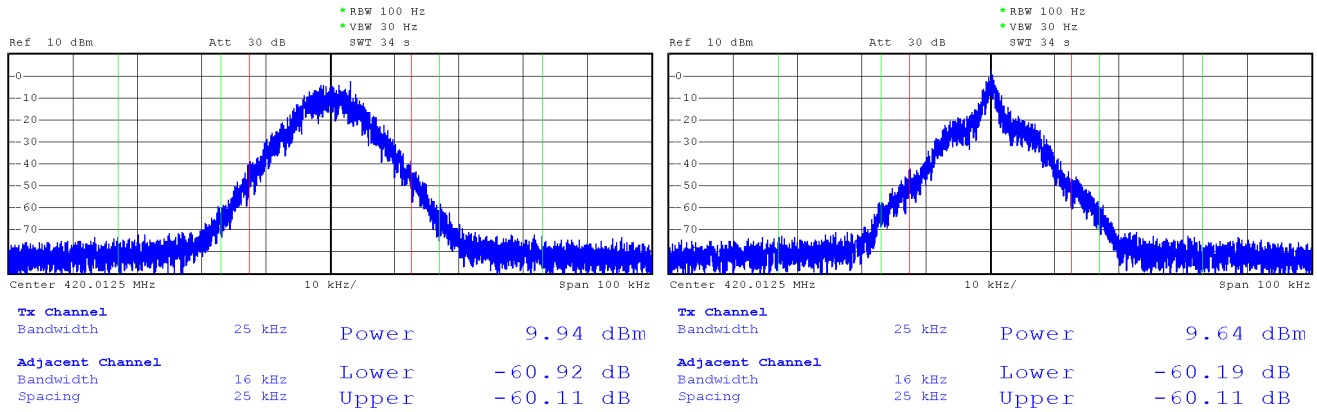


*Fig. 3.2: Modulated signal spectrums. (**left**) 4CPFSK with R=10.4 kBaud, modulation index h~0.3. (**right**) 4CPFSK with R=17.3 kBaud, modulation index h~0.1.*

3.1.3. Transmitter power efficiency

In this section, the measurement results concerning the overall narrowband transmitter power efficiency are presented. It is no ambition however, to provide exact power efficiency analysis of the particular high power amplifier with the selected linearization circuit proceeded. It is rather to give the example of the practically achievable overall transmitter power efficiencies and to show the differences related to selected digital modulation formats of each of the linear/nonlinear class.

[1] The standard [2] specifying the conformity testing for TETRA-like devices allows -55 dBc in normal or -50 dBc in extreme temperature conditions, assuming channel separation of 25 kHz.

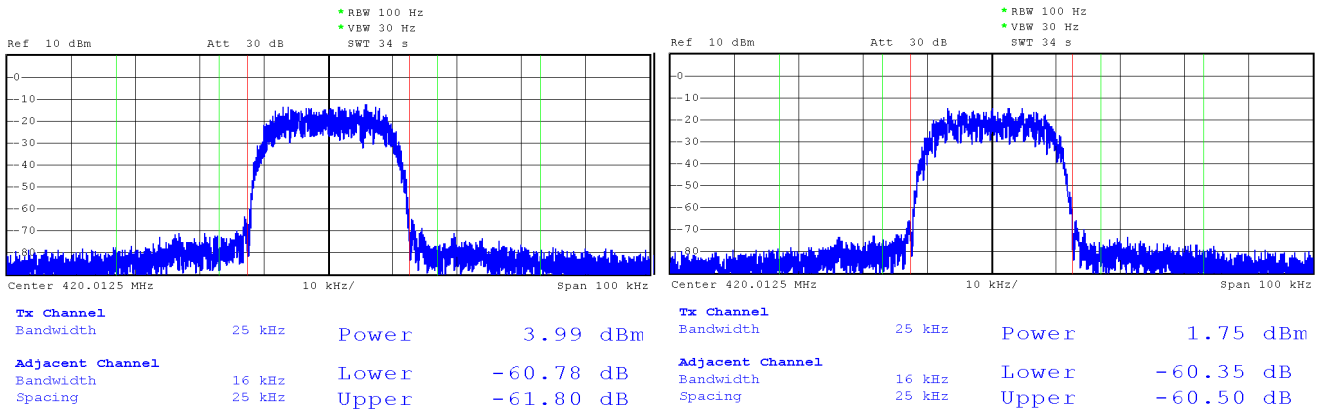


*Fig. 3.3: Modulated signal spectrums. (**left**) π/4-DQPSK with R=17.3 kBaud, (**right**) 16-DEQAM with R=17.3 kBaud.*

As for the linear modulation techniques, the differentially encoded formats π/4-DQPSK, D8PSK and 16-DEQAM have been selected and tested mainly due to their low modulation envelope variations and inherent robustness against negative effects of signal propagation through the narrowband radio channel.

The 2CPFSK and 4CPFSK have been selected from the nonlinear modulation class. There is one particular parameter of high importance essentially influencing the characteristics of these modulation formats and that is a *modulation index*. It expresses the relation between the modulation rate and the maximum frequency deviation according to simple rule (1.1)

$$h = \frac{2\Delta f}{R(M-1)}, \qquad (1.1)$$

where *R* is the modulation rate, *M* is the number of modulation states and *Δf* is the maximum frequency deviation representing the outermost symbol frequency position. The selection of the modulation index in most practical applications of narrowband LMR has been driven by compromising requirements between the modulation rate, receiver sensitivity and adjacent channel power level. Its value usually converges to *1/M* with a well known example of MSK, particularly GMSK where M=2, thus h=0.5 as the lowest value needed to maintain an orthogonal signaling. In order to compare the modulation formats at the same spectrum efficiency we also measured the properties of 2CPFSK and 4CPFSK modulations with very low modulation index resulting in use of high symbol rate of 17.3 kBaud.

The examples of transmitted signal spectrums can be seen in Figure. 3.1 to Figure. 3.3. It is interesting to note the degradation of the signal spectrum with increased symbol rate in case of 2CPFSK and 4CPFSK that implicitly points out that the assigned bandwidth is not used effectively. It can be seen that the significant amount of the signal power is concentrated within the close vicinity of the carrier frequency and thus it results in poor ratio between the occupied signal bandwidth and the noise bandwidth of radio receiver (Table 3.1).

Tab. 3.1: Measurement results of the transmitter parameters for selected modes of operation.

Modulation Format	Symbol Rate	Modul. Parameter	P_{out}	ACI Lower	ACI Upper	Occupied Bandwidth @ 99.9%	P_{IN}	η_{TX}	Spectrum plot
[-]	[kBaud]	[-]	[dBm]	[dBc]	[dBc]	[kHz]	[W]	[%]	[-]
2CPFSK	10.4	h=0.6, α=0.28	40	-62	-61	19.8	35	29	Fig. 3.1
	17.3	h=0.2, α=0.28	40	-62	-61	16.6	35	29	Fig. 3.1
4CPFSK	10.4	h=0.3, α=0.28	40	-61	-60	19.6	35	29	Fig. 3.2
	17.3	h=0.1, α=0.28	40	-61	-60	17.2	35	29	Fig. 3.2
π/4-DQPSK	17.3	α=0.4	35	-61	-62	22.0	22.8	14	Fig. 3.3
D8PSK	17.3	α=0.4	35	-61	-61	22.0	22.8	14	-
16-DEQAM	17.3	α=0.4	35	-60.5	-60.5	22.0	20.4	10	Fig. 3.3
Measurement uncertainty ±2 dB									

The measurement values of achievable output power P_{out}, amount of adjacent channel interference *ACI* and overall transmitter power efficiency η_{TX} are collectively given for all the modulation formats in Table 3.1. It can be seen that the ACI limit (-60 dBc) is maintained for all of these settings; however, there are two penalties in case of linear modulation schemes that typically have to be paid for higher spectrum efficiency. Firstly, it is the lower output power level achievable. For this specific transmitter architecture it is in particular 35 dBm @ π/4-DQPSK, D8PSK and 33 dBm @ 16-DEQAM. Secondly, it is the lower value of the overall transmitter power efficiency reached. Comparing to exponential modes of system operation the efficiency of linear operational modes has decreased to 14% and 10%. Despite this negative trend, the achieved values of output power exceeding 3 W, and 2 W respectively, are considered practically applicable for next generation of narrowband LMR devices and as it will be shown in the next section they enable the system to use its occupied bandwidth with even higher communication efficiency.

3.2. Narrowband radio receiver

The most important parameters which describe the quality of narrowband radio receiver are *maximum usable (data) sensitivity, co-channel rejection, adjacent channel selectivity, desensitization* and *intermodulation response rejection*. Besides the maximum usable sensitivity, all other receiver parameters can be classified as the measures of the *receiver degradation parameters* used to analyze the degradation of its performance due to the presence of unwanted (interfering) signals. Although there is a strong relation between all of these parameters, in this paper the attention is given only to the first of them, to the maximum usable sensitivity in particular.

According to [1], the maximum usable data sensitivity is the minimum level of the signal (emf) at the receiver input, produced by a carrier at the nominal frequency of the receiver, modulated with a normal test signal, which will, *without interference*, produce, after demodulation, a data signal with a specified *bit-error-ratio* (BER) of 10^{-2} or a specified *successful message ratio* (SMR) of 80%.

*The maximum usable sensitivity shall not exceed an electromotive force of **3.0 dBμV** under normal test conditions.*

Assigning this value as *S*, one can also express what *signal-to-noise ratio* (SNR) can be expected in relation to *noise figure* (NF) and transformed to the receiver input

$$SNR = S - (10.\log(kT) + 10.\log(B_N) + NF) \quad [dB]. \qquad (2.1)$$

In (2.1), *k* is the Boltzmann's constant, *T* is the absolute temperature in Kelvin and B_N is the receiver noise bandwidth of e.g. 25 kHz.

3.2.1. Maximum usable data sensitivity

In this section, the results of maximum usable data sensitivity measurement (Figure 3.4) for the complete narrowband radio transceiver are presented. All the results are given for 25 kHz channel separation.

Firstly, let us focus on operational modes with exponential modulations, Figure 3.4. It can be seen that the *emf* sensitivity limit of +3 dBμV (-110 dBm @ 50 Ω) is fulfilled with margin for both modulations (2CPFSK, 4CPFSK) when running at the symbol rate of 10.4 kBaud. When higher symbol rates are selected, these modulations loss their power efficiency rapidly and for the selected symbol rate of 17.3 kBaud, the sensitivities lower down to the values of −107 dBm @ BER=10^{-2} and −102 dBm @ BER=10^{-2} for 2CPFSK and 4CPFSK respectively. This discrepancy is caused mainly due to the fact that there is a significantly lower frequency deviation used at the higher symbol rates. The decrease in power efficiency with increasing spectrum efficiency is not linear as for the typical linear modulations. Although possible, this example documents that the increase in spectrum efficiency of exponential modulation techniques cannot be considered for efficient use of assigned bandwidth.

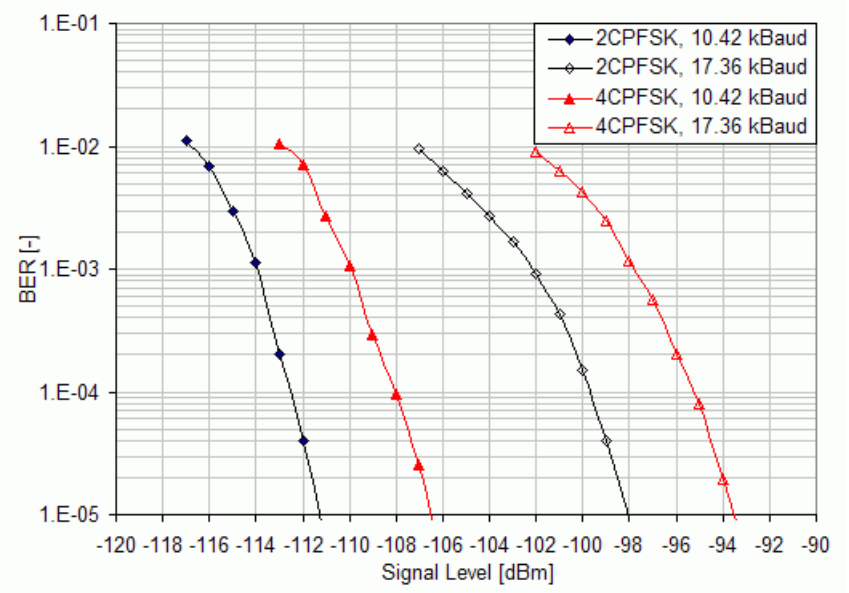


Fig. 3.4: Maximum usable sensitivity measurement results for different settings of exponential modulations.

The second set of measurement results, presented in Figure 3.5, documents the power efficiency analysis of operational modes based on the linear modulation techniques. It can be seen that when using the linear π/4-DQPSK, the radio receiver can still reach the data sensitivity limit even for 17.3 kBaud with a 2 dB margin. Even from this comparison it is evident that the π/4-DQPSK mode of operation outperforms the 4-CPFSK at higher spectrum efficiencies. Further increase in spectrum efficiency can be reached by higher order constellations such as D8PSK and 16DEQAM and the radio receiver can still maintain practically applicable sensitivities of −107 dBm @ BER=10^{-2} and −105 dBm @ BER=10^{-2} respectively.

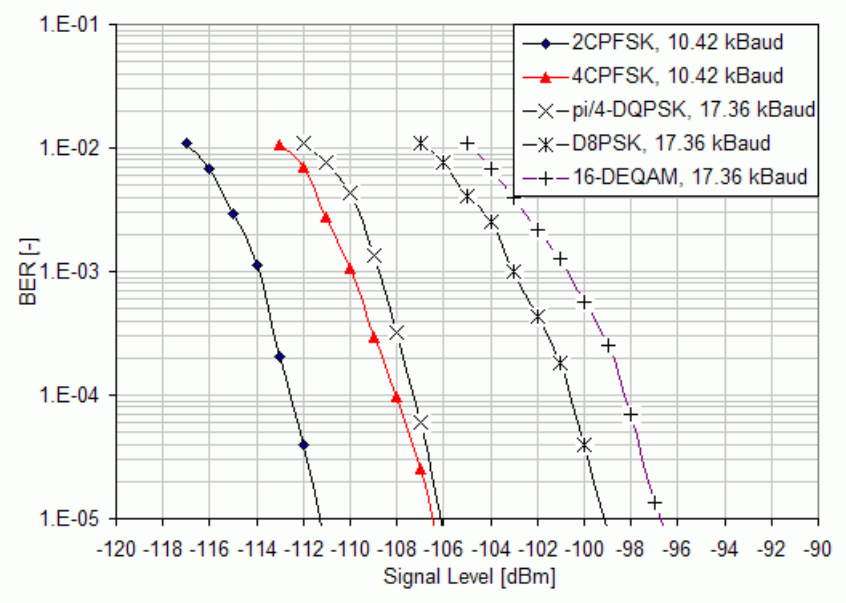


Fig. 3.5: Maximum usable sensitivity measurement results. Channel separation 25 kHz.

3.2.2. Efficient use of narrowband radio channel

As it has been written in the Section 1, the radio transceiver in exponential modulation mode can make use of higher transmitter power. In order to take this fact into account the *system gain (SG)* or the *maximum allowed path loss* (2.2)

$$SG\ [dB] = P_{out} - S\ , \qquad (2.2)$$

is usually calculated for the wireless communication systems. Here the P_{out} is the available transmitter power expressed in dBm and S is the measured value of radio receiver sensitivity, also in dBm. It expresses the referential value of the link budget, assuming 0 dBi of antennas gain and together with the *spectrum efficiency* given by (2.3) it expresses how effectively the radio device uses its assigned bandwidth

$$\eta\ [bit/s/Hz] = \frac{R_b}{B}\ . \qquad (2.3)$$

In (2.3), the R_b is the *raw bit rate* given in [bits/s] and B is the frequency bandwidth assigned to the radio system, 25 kHz in particular.

All these performance characteristics are collectively given in Table 3.2. It can be seen that even with the lower available transmitter power, the radio transceiver can reach wider system gain at higher spectrum efficiencies while running in linear as oppose to the exponential modulation mode. On the other hand, if the long distance coverage is of the primary application concern, even the 2CPFSK modulation having spectrum efficiency of 0.4 bit/s/Hz, but the system gain of impressive, 157 dB, can be a reasonable option.

Tab. 3.2: Overall performance characteristics of the narrowband radio transceiver for selected modes of operation.

Modulation Format	Modul. Param.	Symbol Rate	Raw Bit Rate	Spectrum Efficiency	Data Sensitivity @ BER 10^{-2}	Available Output Power	System Gain
[-]	[-]	[kBaud]	[kbits/s]	[bit/s/Hz]	[dBm]	[dBm]	[dB]
2CPFSK	h=0.6, α=0.28	10.42	10.42	0.42	-117	40	157
	h=0.2, α=0.28	17.36	17.36	0.69	-107	40	147
4CPFSK	h=0.3, α=0.28	10.42	20.83	0.83	-113	40	153
	h=0.1, α=0.28	17.36	34.72	1.39	-102	40	142
π/4-DQPSK	α=0.4	17.36	34.72	1.39	-112	35	147
D8PSK	α=0.4	17.36	52.08	2.08	-107	35	142
16-DEQAM	α=0.4	17.36	69.44	2.78	-105	33	138
Measurement uncertainty ±2 dB							

3.3. Conclusion

As it was shown in this paper, the strict limits of the referenced standard as well as the state of the technology hindered increasing the communication efficiency with which the narrowband systems have been using the occupied frequency bandwidth. The key limiting factor that has been identified was the limit of adjacent channel power attenuation. Lessening the requirement from -70 dBc to -60 dBc in 2007 has opened up the closed door for implementation of linear digital modulation techniques. However, as it has been shown in later sections, a reasonable use of the exponential modulation can be still beneficial for these systems. Based on the results presented, the most important concluding notes can be seen in the following:

- When the long distance coverage as well as the overall power efficiency are of the primary application concern, the use of exponential modulation techniques 2CPFSK and 4CPFSK at relatively low symbol rates e.g 10.4 kBaud can be the recommended option. In this case, the nonlinear modulation techniques can make use of higher frequency deviation and increase the system gain by outstanding values of receiver sensitivities. At the 10 W of output power the system gain of 157 dB and 153 dB for 2CPFSK and 4CPFSK modulation techniques respectively can be expected.

- When higher symbol rates are selected, the exponential modulation techniques lose their power efficiency (and their main advantage) significantly. Further increase of the exponential modulation spectrum efficiency from the values currently being used by the narrowband systems (up to 1 bit/s/Hz) can be therefore considered inefficient.

- From all the modulation formats studied, the π/4-DQPSK can provide the narrowband LMR system with communication efficiency closest to the optimal communication systems. The proposed solution based on this modulation technique can reach the spectrum efficiency of up to 1.5 bit/s/Hz. The data sensitivity limit required by [1] can also by fulfilled with margin of 2-3 dB, resulting in the system gain of 147 dB.

- For applications where higher data throughputs are needed the additional increase in spectrum efficiency can be gained by D8PSK and 16-DEQAM modulation formats. However, compared to π/4-DQPSK, an increase in overall communication efficiency cannot be expected, while there is the inevitable penalty in power efficiency characteristic.

References

[1] ETSI EN 300 113-1 V1.6.2 (2009-11), Electromagnetic compatibility and Radio spectrum Matters (ERM), Part 1: Technical characteristics and methods of measurement. European Standard. ETSI, 11/2009.

[2] ETSI EN 302 561 V1.2.1 (2009-12), Electromagnetic compatibility and Radio spectrum Matters (ERM), Land Mobile Service; Radio Equipment using constant or non-constant envelope modulation operating in a channel bandwidth of 25 kHz, 50 kHz, 100 kHz or 150 kHz; Harmonized EN covering essential requirements of article 3.2 of the R&TTE Directive. European Standard. ETSI, 12/2009.

[3] ETSI EN 301 166-1 V1.3.2 (2009-11), Electromagnetic compatibility and Radio spectrum Matters (ERM), Part 1: Technical characteristics and methods of measurement. European Standard. ETSI, 11/2009.

4. Autospeed

Normally all radio modems in a network have to transmit with the same data rate on the same radio channel. The Autospeed feature of RipEX enables different speeds to be used simultaneously in a radio modem network.

The following picture gives an example of a network layout. Let us assume, that all signals are strong enough to ensure almost perfect operation:

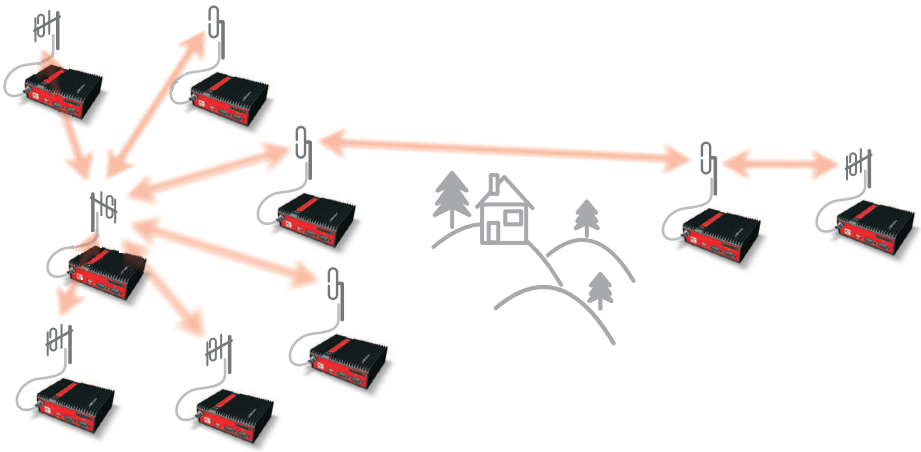


Fig. 4.1: Autospeed - initial situation

After some time situation changes and path loss on one of these links significantly increases, rendering the communication unreliable:

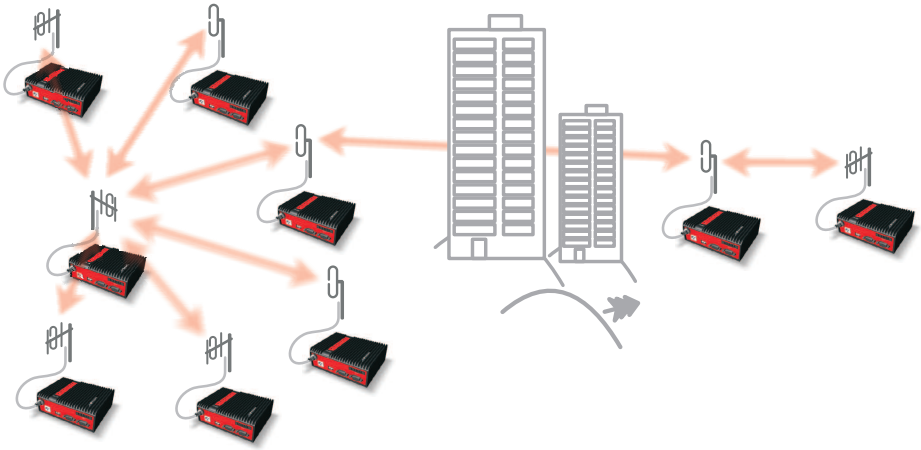


Fig. 4.2: Autospeed - problem

What can we do:

- Change antennas on one or both sides of the link
- Use higher masts on one or both sides of the link
- Build additional repeater(s)
- Lower the data rate significantly to increase the system gain

The first three possibilities require time and money, i.e. additional investment. The fourth possibility (when applied to whole network, as it normally is the case) would slow down the response time (two

to four times) of the whole network, quite probably making it unusable for the application. RipEX Autospeed feature allows to change the transmission data rate at the affected radios only, the rest of the network may continue in full speed. Consequently the overall performance of network is maintained practically at the same level while no additional investment is required. More over, the whole fix can be done in minutes from behind a web-browser screen while sitting in your office.

Of course a similar scenario can be used right from the moment of planning a new network. The investment cost can be reduced by purposefully configuring the few „difficult“ radio links to a lower data rate.

The above scenarios are made possible by the unique capability of RipEX to automatically adjust its receiver to the data rate of the incoming frame. Note that when an ACK frame is sent by the receiving RipEX, it always uses the same data rate as the frame it acknowledges. The only limitation of this feature is that all the frames have to have the same symbol rate and the same principle of modulation (i.e. CPFSK or linear).

Modulation types which can be combined within one approval type (FCC or CE):

2CPFSK & 4CPFSK with or without FEC
or
D2PSK & Pi/4DQPSK & D8PSK &16DEQAM with or without FEC

The improvement in system gain value using this technique may be more than 15 dB. Increasing gain of antenna system by that value would be impractical, often impossible – the „difficult“ hops are designed to use high-gain directional antennas from the beginning. Hence the Autospeed may make a radio modem network the optimum choice in situations where it could not be economically feasible before.

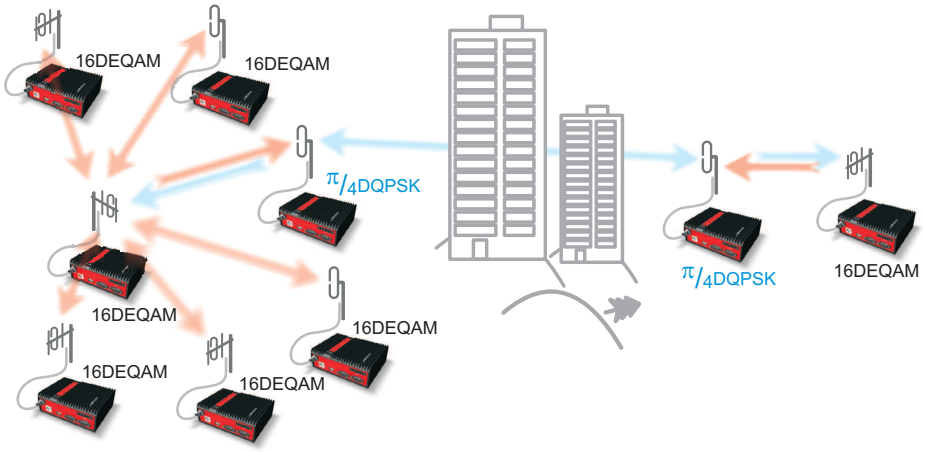


Fig. 4.3: Autospeed - solution

5. Back-to-Back repeater

This layout and settings may be used if you need to operate different parts of the radio network on different frequencies. Connection between these two parts is realised by Back2Back connection between two RipEX's (hereafter referred to as border RipEX's), each of which operates on different frequency.

5.1. Back to Back in Bridge mode

Ethernet

If end devices are connected to RipEX's over Ethernet, border RipEX's can be connected with an Ethernet cable. IP addresses of all RipEX's as well as connected devices must be within the same LAN. Ethernet interfaces must be interconnected for proper function of remote service access.

COM

If end devices are connected to RipEX's over COM interface, one (any of the two) COM port of a border RipEX must be connected to a COM port of the other border RipEX using RS232 crossover cable or null modem. Communication parameters of both connected ports must be set to the same values, we recommend using the highest available speed.

Important

Border RipEX's should be interconnected via one COM port only, connecting both COM ports would create a loop.

Limitation: If a device is connected to the free COM port of a border RipEX, it only sends data to its part of the radio network. Data from all other COM ports of other RipEX's throughout the entire network will be delivered to both COM ports of all other RipEX's.

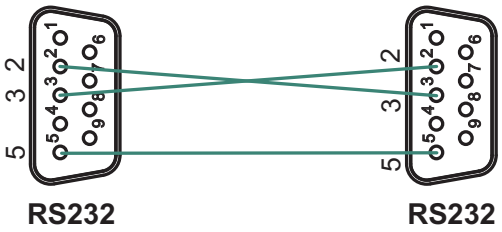


Fig. 5.1: Crosslink serial cable

Ethernet + COM

If end devices are connect to RipEX's both over Ethernet and COM ports, or if you require remote access to a network which uses COM ports, border RipEX's must be interconnected both via Ethernet (see 1.1) and COM (see 1.2).

5.2. Back to Back in Router mode

In Router mode border RipEX's are interconnected by Ethernet cable. Routing in both parts of the network must be set up so that communication passes through the Ethernet interface of the border RipEX's. We recommend splitting both radio networks to two separate LAN networks.

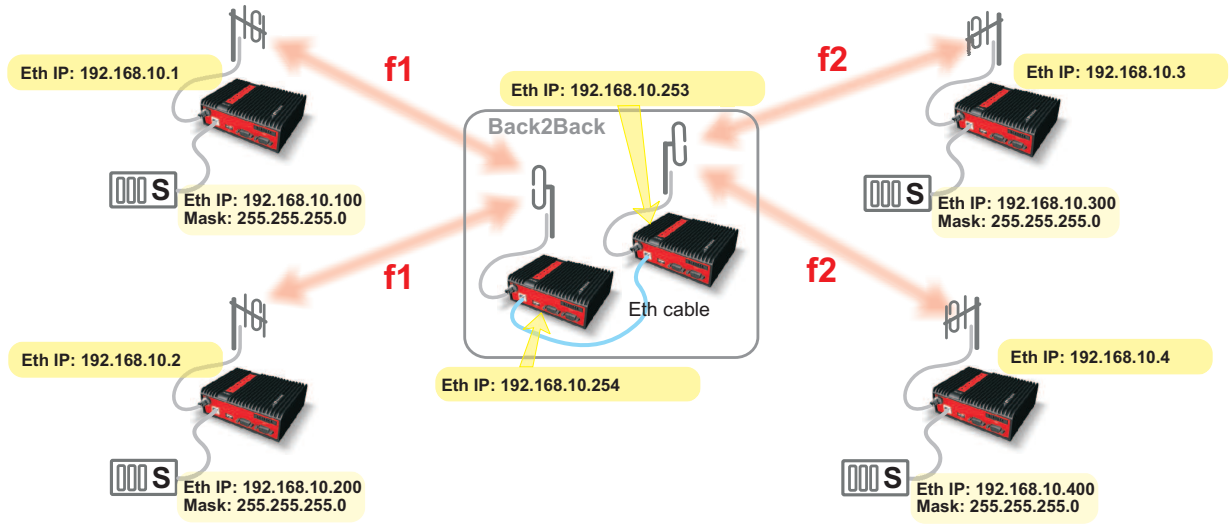


Fig. 5.2: Back2Back in bridge mode

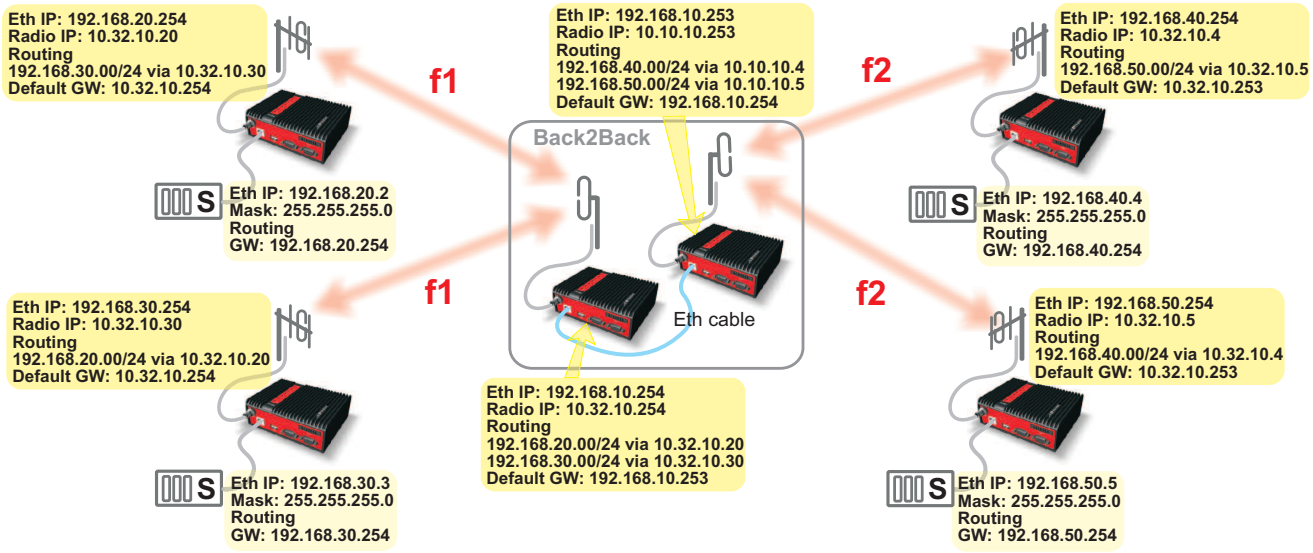


Fig. 5.3: Back2Back in router mode

6. Combining MORSE and RipEX networks

When expanding a MORSE network with RipEX radio modems, different arrangements are possible. In the following paragraphs we assume that the whole network is divided into two parts – the MORSE part and the RipEX part. The two parts are interconnected through two radio modems – one MRxxx and one RipEX, hereafter referred to as border radio modems. As RipEX and MRxxx radio channel protocols are not compatible, we strongly recommend you use different frequencies for either part of the network.

6.1. RipEX part in Bridge mode

There are two basic scenarios:

- Terminal devices are connected to Ethernet interface
- Terminal devices are connected to COM port

6.1.1. Terminal devices connected over Ethernet

If terminal devices are connected over Ethernet, the border RipEX and MRxxx should also be interconnected by an Ethernet cable. The IP addresses of all devices in the network should belong to a single LAN.

The picture shows MORSE network settings; note the use of Proxy ARP in IP-M-IP mode.

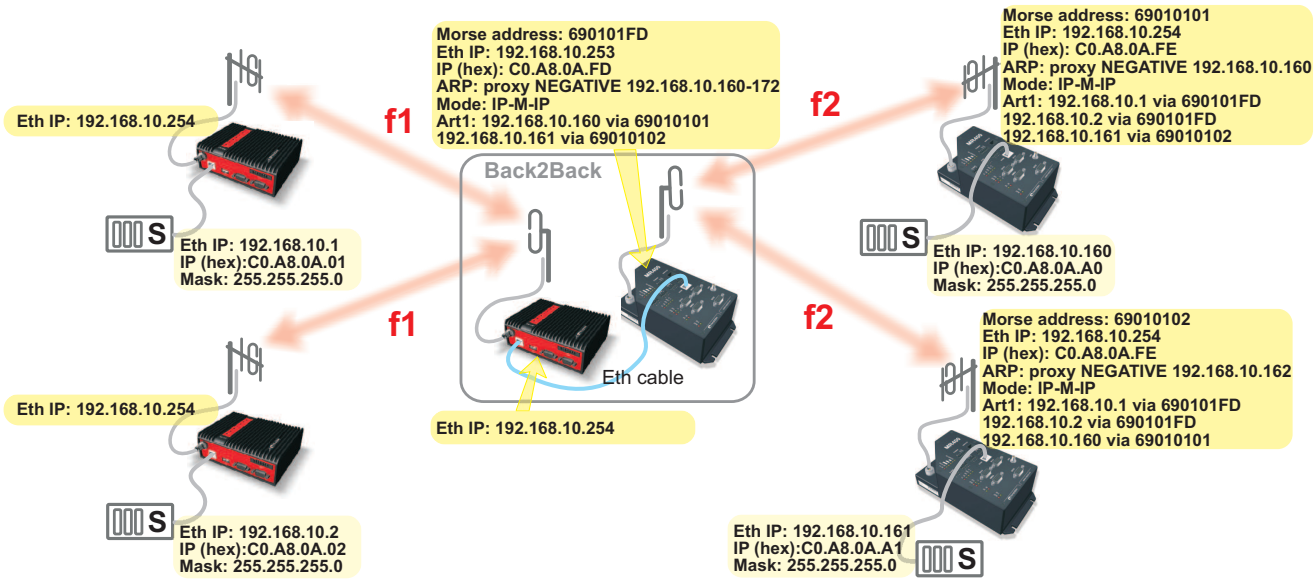


Fig. 6.1: RipEX - MR400 in Bridge mode

6.1.2. Terminal devices connected to COM

The COM port of the border RipEX and the RS232 of the border MRxxx are connected with a crosslink serial cable, see Fig. 6.2, “Crosslink serial cable”.

The COM port protocol at the border MRxxx must be the same as protocol used by the other MORSE devices in the network. In some special cases, the ASYNC LINK protocol can be used for the border interconnection.

If the Master is located on the side of the MRxxx, the border MRxxx should be set to Slave. Depending on the SCC interface used the MRxxx should use Multiaddressing with addresses of all the Slave units on the RipEX network.

If the Master is located on the side of the RipEX, the border MRxxx is set like it was connected to the Master and the Node of the connected SCC interface has to correspond to the Master's address.

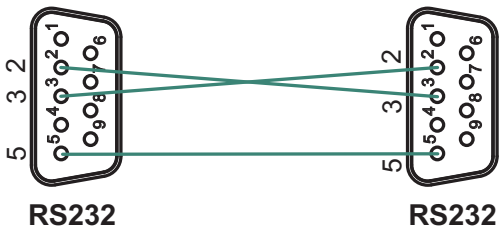


Fig. 6.2: Crosslink serial cable

6.2. RipEX in Router mode

There are two basic scenarios:

- Terminal devices are connected over Ethernet
- Terminal devices are connected over COM interface

6.2.1. Terminal devices connected over Ethernet

In this scenario the border RipEX and MRxxx should be interconnected with an Ethernet cable.

Routing in both parts of the network should be set up so that communication between them is channeled over the border modems. It is recommended that terminal devices in the two parts of the network are located on separate LAN's.

The picture shows MORSE network settings.

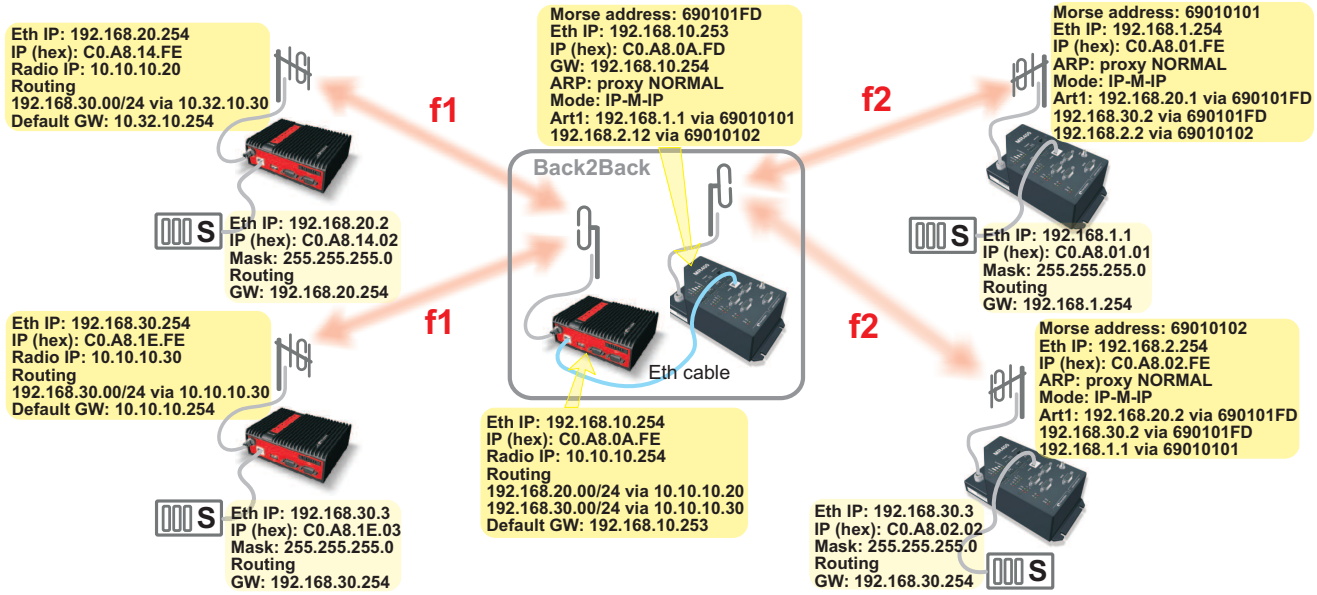


Fig. 6.3: RipEX - MR400 in Router mode

6.2.2. Terminal devices connected to COM

A MORSE network can only be expanded with RipEX modems if the application protocol is supported both by MORSE and RipEX, or if RipEX's UNI protocol can be used instead. If you want to use protocols which are not implemented in RipEX by default, please consult RACOM's technical support.

The COM port of the border RipEX and the RS232 of the border MRxxx are connected with crosslink serial cable, see Fig. 6.2, “Crosslink serial cable”.

If the Master is located in the MORSE part of the network, the border MRxxx should use Multiaddressing for addresses of all Slaves in the RipEX network. Protocol settings should reflect that. The border RipEX then should be set up as connected to the Master using the appropriate protocol (address translation using a mask or table, routing rules).

If the Master unit is located on the RipEX side of the network, rules for address translation should direct all the packets sent to Slave units of the MORSE network to the COM port connected to the border MRxxx. This COM port should then use an appropriate protocol in Slave mode. In the border RipEX the timeout for response from technology should be extended from 500 ms to several seconds (the response time will depend on the size of the MORSE network) – this parameter can only be set in CLI. On the MORSE side, the protocol should be set to Master.

7. Profibus

Radio modem RipEX supports the most widely spread Profibus (Process Field Bus) type designated Profibus DP (Decentralized Periphery) type 0 (see http://www.profibus.com/technology/profibus/).

Profibus DP is designed for fast master–slave communication. The central master unit communicates with the remote slaves using RS485 bus. They are typically connected by twisted pair cabling. The cable length between two RS485 repeaters is limited (from 100 to 1200 m), depending on the bit rate used. The RipEX Profibus DP implementation allows for RS485 to be replaced by radio network, either partially or entirely. This significantly increases the potential distance between the individual nodes or even enables you to get rid of cable links altogether.

7.1. Bridge and Router modes

RipEX operates in two basic modes, Bridge and Router. Network topology determines which one is the more suitable for your specific application (see chapter RipEX in detail[1] of the manual).

Apart from network layouts designed in this manual, we also recommend using Router mode if alongside the central RipEX some PLC Slaves are also connected to the PLC Master over RS485 while others connect over the radio network.

This is because in Bridge mode RipEX would broadcasts to radio channel each packet received on RS485. This could cause slower communication in some situations, and even collisions when a repeater is used. In Router mode only the packets destined for remote PLC Slaves are broadcast over the radio channel while packets sent to the PLC Slaves connected directly over RS485 are ignored.

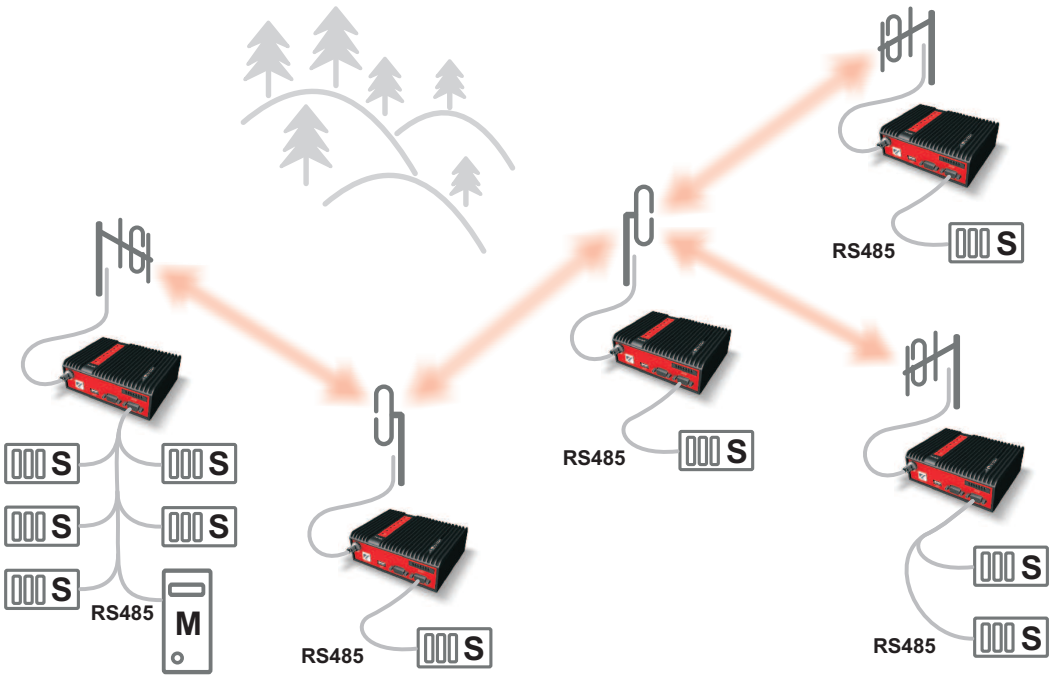


Fig. 7.1: RS485 and Radio network

[1] http://www.racom.eu/eng/products/m/ripex/ripex-detail.html

7.2. Profibus settings

We will only be looking at the basic communication parameters of the protocol – other parameters correspond to the standard Profibus DPV0. Profibus protocol is very sensitive to DP Slave response times. Delays are common in radio networks; this should be taken into account when setting up Profibus communication parameters.

Recommended default Profibus settings for data transfer using RipEX radio modems:

Tslot_Init:	16 383 t_bit
Max. Tsdr:	50 t_bit
Min. Tsdr:	11 t_bit
Tset:	1 t_bit
Tqui:	0 t_bit

Explanation of acronyms:

Tslot_init (Slot-time): This indicates how long a DP Master should wait for a response from a DP Slave before it repeats a packet or sends another. The maximum value is 16 383.

Max. Tsdr (Maximum Station Delay of Responders): Sets the maximum DP Slave response time. This value is the same for all DP Slaves and is distributed from the DP Master at the beginning of their communication. This value must be lower than Tslot_init (Slot-time).

Min. Tsdr: Sets the minimum DP Slave response time. 11 to 255 bit values are permitted. This value is the same for all DP Slaves and is distributed from the DP Master at the beginning of their communication. This value must be lower than Max. Tsdr.

Tset: Sets delay. This is used to postpone broadcasting of the next packet. This parameter enables you to create space for other communication on RipEX network.

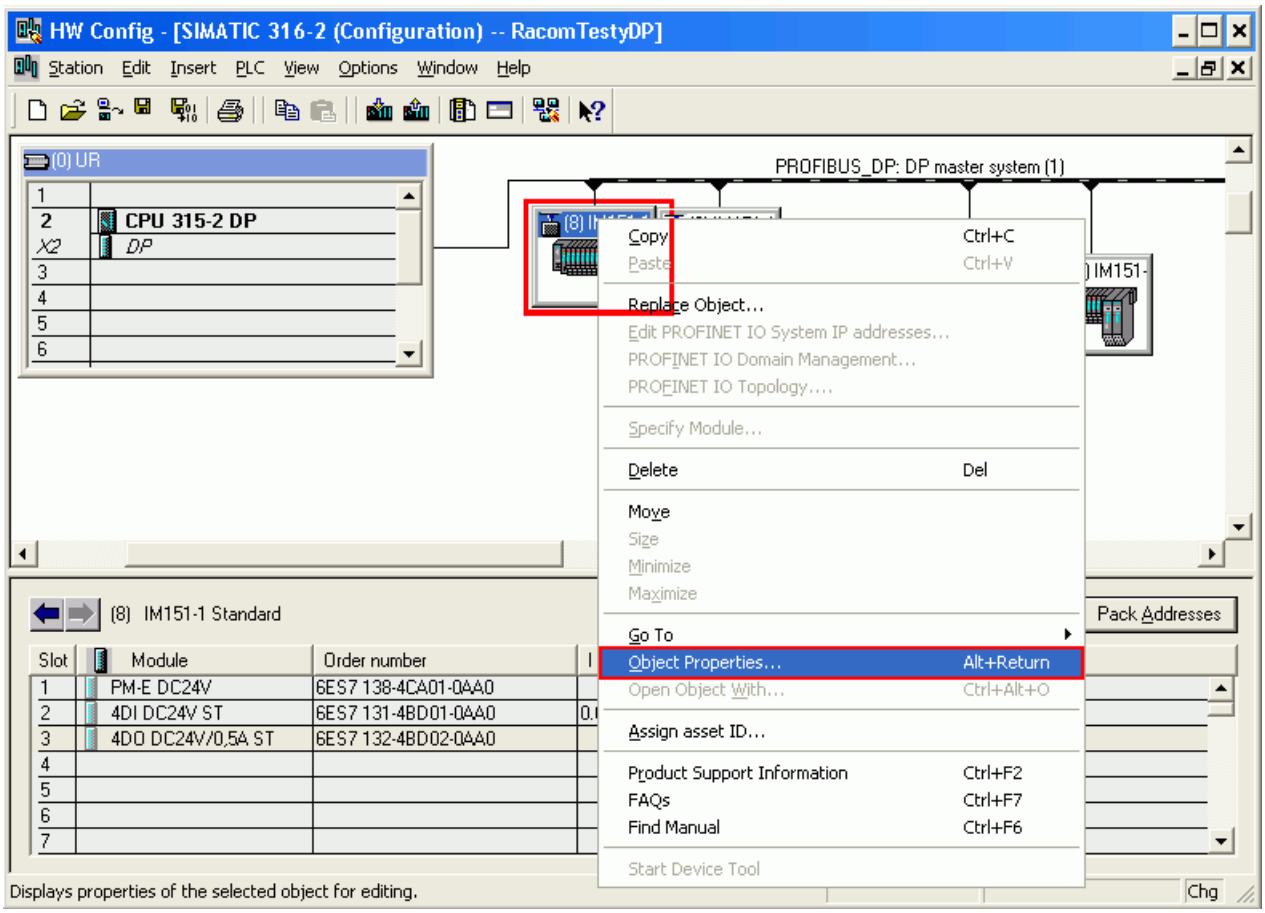
Tqui (Quit time): Sets the switching time between reception and broadcasting. This must be lower than Min. Tsdr.

Note: All times are given in bits. 1 t_bit = 1 / Baud rate [seconds]

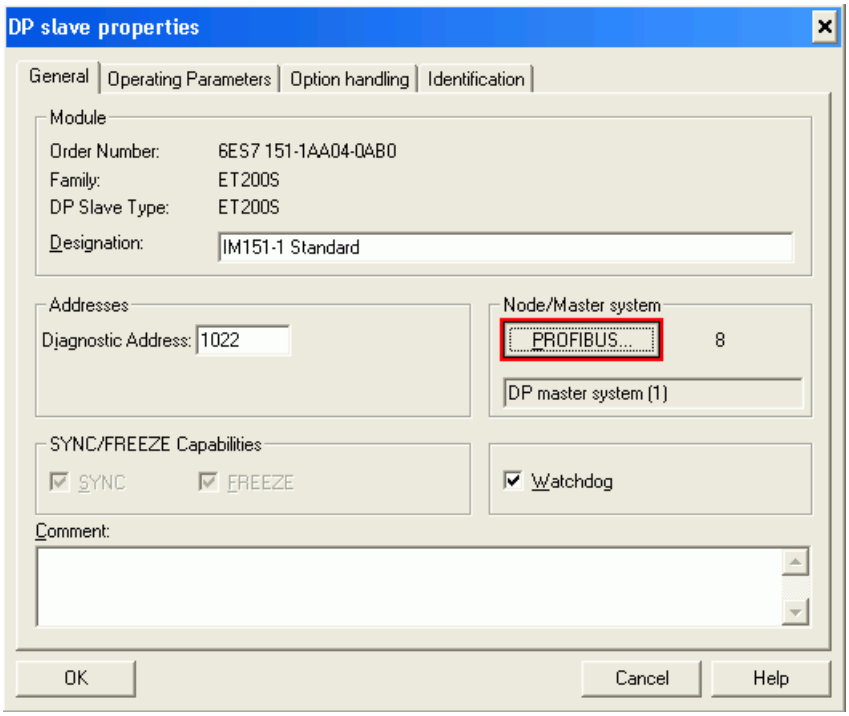
A single bit time	Baud rate – data transfer speed
104.2 µs	9600 bps
52.1 µs	19200 bps

Example of Profibus DP settings in STEP 7

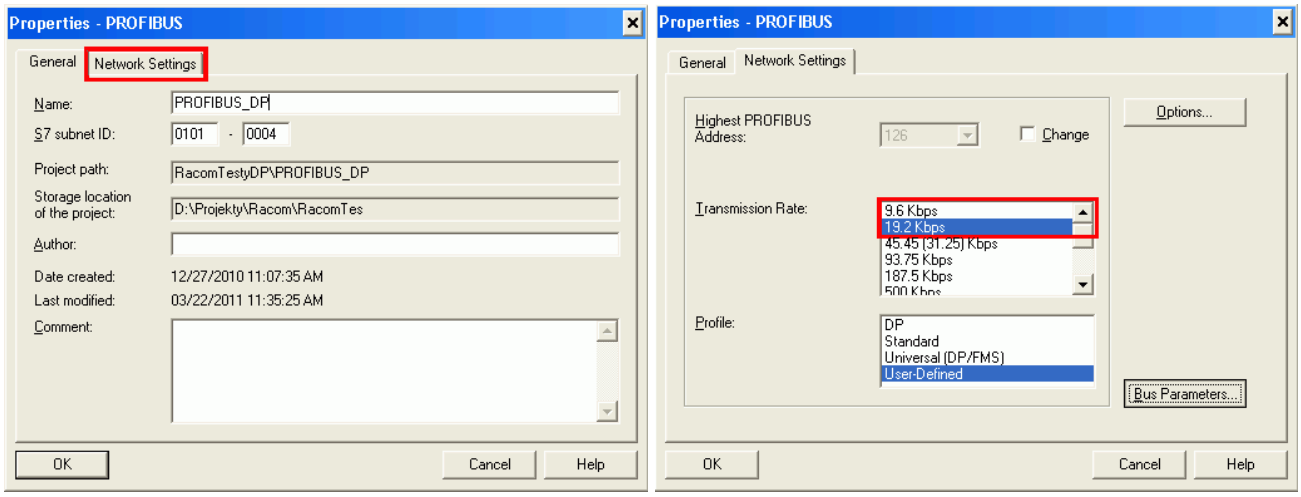
Under network layout click the right mouse button to open Object Properties:



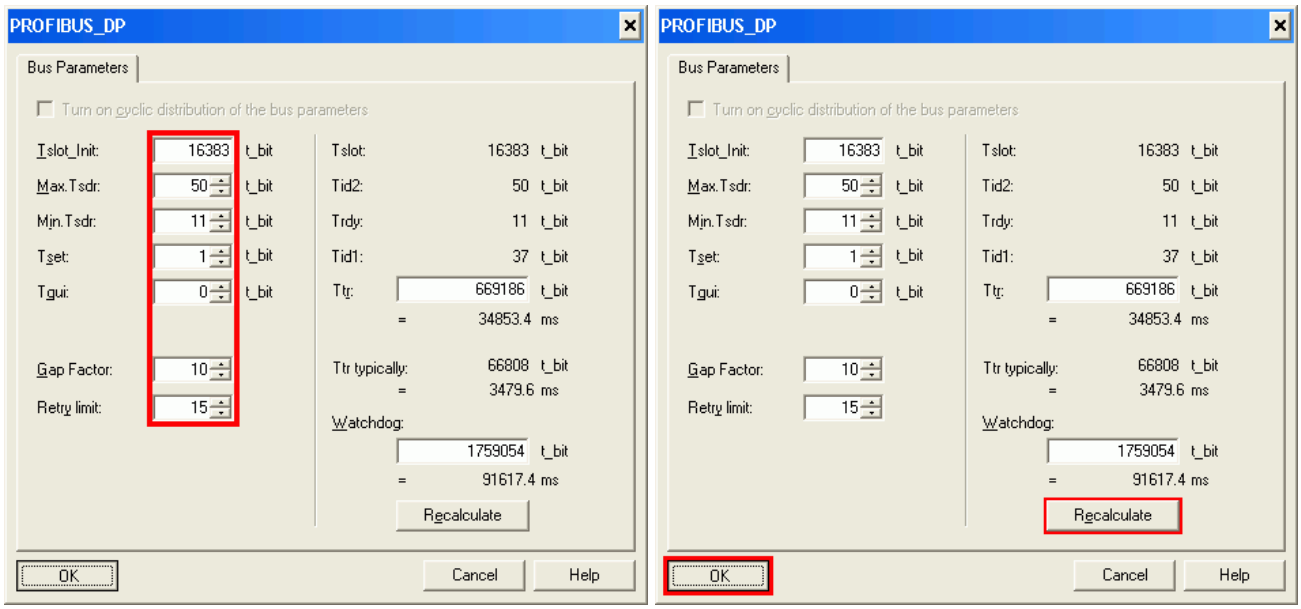
DP slave properties window opens. Click on the PROFIBUS button:



Properties – PROFIBUS window opens. Select the Transmission Rate (19.2 Kbps or 9.6 Kbps) under the Network Settings tab. The recommended value is 19.2 Kbps. Under Profile select User Defined and click Bus Parameters.



PROFIBUS_DP is the most important settings window; fill in settings as shown below, click Recalculate and confirm by clicking OK. Confirm the values in all open windows and click the icon Download to Module. Tslot_Init is a value which fundamentally influences operation of the entire device. 16 383 t_bit is the maximum value which helps test radio transmission. We recommend setting as described in chapter "Advanced Settings – Calculation of minimum slot time".



7.3. RipEX settings

7.3.1. Operating mode

See chapter Advanced configuration[2] of the manual.

If there is no more than a single repeater on your network, we recommend using Bridge mode. Profibus DP is always a master–slave type network in which there is no danger of radio channel collisions.

[2] http://www.racom.eu/eng/products/m/ripex/h-menu.html

Router mode should only be used where network topology does not allow for Bridge mode to be used (see page YY of the manual). If you choose to use Router mode we recommend switching off acknowledgement on the radio channel. This speeds up packet transmission on the radio channel. Repetition of undelivered packets is ensured through the application layer of the DP Master.

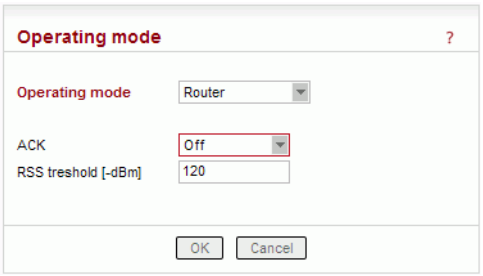


Fig. 7.2: ACK Off

7.3.2. COM 2

Profibus DP utilises RS485 interface. This interface can only be set to COM2 in RipEX. COM2 functionality is conditioned by using the appropriate software key, see chapter Maintenance[3] of the manual.

COM2 settings must correspond to PLC device settings. We recommend setting port speed to 9600 for complex networks or 19200 bps for networks without re-translation (the timing is derived from the length of a single bit).

Idle state can be reduced to as little as 1.

In Router mode, set Protocol to Profibus.

For explanation of the individual parameters refer to on-line help in the web interface or chapter Settings[4] of the manual.

Note: If Profibus IP's do not correspond to RipEX IP's (e.g. several PLC Slaves are connected to a RipEX over a single bus), addresses must be **translated using a table**.

7.4. Advanced settings

7.4.1. Calculation of minimum slot time

Setting the appropriate (minimum) Tslot_Init value for a given network may significantly shorten the total DP Slave polling cycle. If one of the DP Slaves is out of order or if its response is lost, the DP Master will only wait for a set minimum time before sending another query. The value should be set to maximum to prevent problems.

The calculator on http://www.racom.eu/eng/products/radio-modem-ripex.html#calculation enables you to calculate the RTT (round trip time).

Set the PLC Master to Ethernet interface in the calculator (Profibus protocol timing is based on the last sent byte; time on Master's RS485 does not figure in this calculation).

RTT for Bridge mode can be used directly; for Router mode the resulting average RTT needs to be multiplied by constant 1.25 to receive the maximum achieved RTT.

Calculate the recommended Tslot_Init as follows:

Tslot_Init = RTT * (Port speed in bps) / 1000

[3] http://www.racom.eu/eng/products/m/ripex/h-menu.html
[4] http://www.racom.eu/eng/products/m/ripex/h-menu.html

7.4.2. Router mode - timing

Router mode web based settings may cause time problems in more complex networks. CLI lets you adjust radio channel access parameters and set up repetition taking into account the number of re-translations in your radio network.

If you only use the Profibus protocol with RipEX and no other broadcast interferes with your network, you can configure certain parameters to shorten the access time to channel using CLI. If you want to use packet acknowledgement on the radio channel, you can shorten the repetition timeout if ACK is turned off.

Set up using CLI:

cli_cnf_set_device_mode:

-ack n	Turns on ACK
-retries 2	Number of retries 2
-rto-prog f	Turns off progressive retries
-rto-fix 10	Shortens the retry timeout to the minimum value of 10 Bytes
-rto-var 10	Shortens the variable retry timeout to the minimum value of 10 Bytes
-slots-rx 0	Will receive immediately after request – random channel access is not used
-slots-tx 0	Will transmit immediately after request – random channel access is not used

Same settings should be used for all devices.

To find out more about CLI, see RipEX manual chapter CLI Configuration[5].

Set the following in Profibus parameters:

Tslot_Init 16383

Note: This setting is only appropriate for certain types of networks; changes should only be made by experienced users!

Connecting RS 485

Connector layout of RipEX COM 2 for RS 485 and the corresponding PIN's on Siemens Simatic S7.

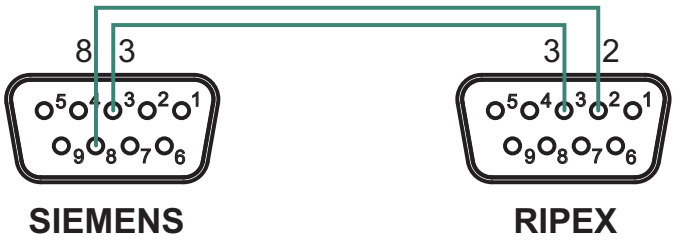


Fig. 7.3: RS485 connection

[5] http://www.racom.eu/eng/products/m/ripex/cli-conf.html

8. Modbus TCP/RTU

Use of Modbus in RipEX.

RipEX supports Modbus RTU, Modbus TCP as well as their combinations:

Tab. 8.1:

	Centre protocol	Remotes' protocol	Radio network behaviour	Available with Operating mode
1	RTU	RTU	Modbus RTU over Radio channel	Bridge, Router
1.1	Multiple Masters RTU	RTU	Modbus RTU over Radio channel	Router
2	TCP	TCP	TCP/IP protocol over Radio channel	Bridge, Router
3	TCP	TCP	TCP/IP protocol locally between Modbus device and RipEX. TCP/IP overhead is not transferred over Radio channel	Router
4	TCP	RTU	Conversion of Modbus TCP to Modbus RTU on the remote units	Router
5	TCP	Combination of TCP and RTU	Using 3 and 4	Router
6	Multiple TCP and multiple RTU masters	Combination of TCP and RTU	TCP master communicates with TCP or RTU slaves, RTU Master only communicates with RTU slaves, utilising 1.1 and 5	Router

8.1. Modbus RTU

A standard simple network design with a single Master and several Slaves running Modbus RTU.

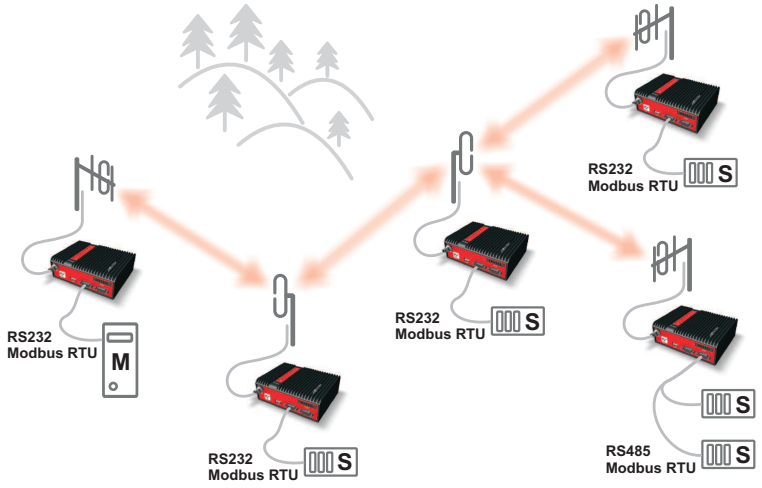


Fig. 8.1: Modbus RTU

In Bridge mode, set the type of communication interface (RS232 or RS485) for the COM port as well as the parameters of the serial interface, both for the Master and Slave.

In Router mode, set the COM port of your Master RipEX to Modbus (Mode of Connected device). To translate Modbus addresses to RipEX format and vice versa either use a mask (if RipEX addresses mirror the Modbus ones) or table. A table must be used if there are several Modbus slaves behind a single RipEX (RS485 or both COM1 and COM2). For more information refer to on-line help or chapter Protocols / Common parameters[1] of the manual.

In addition, set Modbus to Slave on all remote units. If you intend to broadcast in Modbus, set the required parameters. For more information refer to on-line help or chapter Protocols / Slave[2] of the manual.

8.1.1. Modbus RTU with multiple Masters

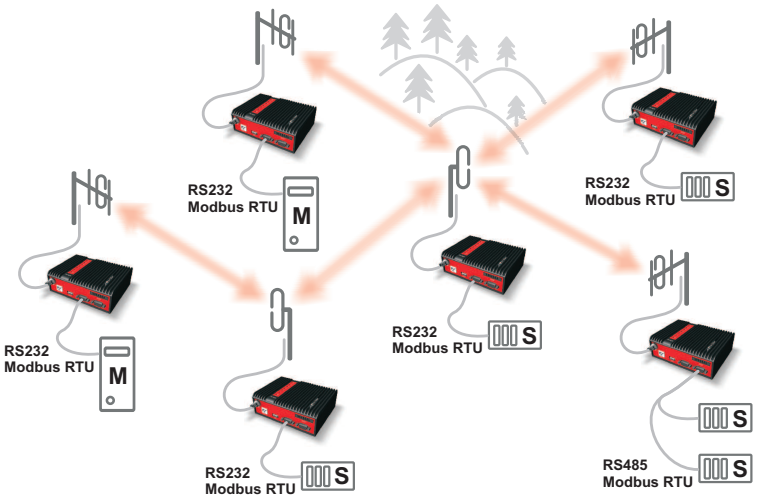


Fig. 8.2: Modbus RTU with multiple Masters

RipEX allows for several Masters to operate at the same time and to communicate with the same Slaves. Router mode is presumed in this design. RipEX settings remain the same as above. Each Slave responds directly to the Master unit which queries it – i.e. if Master A issues a query to a Slave, the response is sent exclusively to Master A. If a single Slave is queried by two Masters at once, queries are resolved one by one. Query from the second Master is queued inside RipEX until it receives a response from Slave RTU on its serial interface or until 500 ms timeout has passed.

8.2. Modbus TCP

A standard simple network with a single Master and several Slaves running Modbus TCP. A TCP/IP connection is established and maintained between Master PLC and Slave RTU across the entire radio network.

In Bridge mode, no special setup is required. RipEX operates as an intelligent Bridge. For more information refer to on-line help or chapter ETH / Modbus TCP[3] of the manual.

In Router mode, routing must be set up in the radio network. Communication between the IP address of the Modbus Master and IP addresses of all Modbus Slaves is necessary. Remember to set Modbus TCP/RTU and Terminal Servers (under Settings/Ethernet) to Off.

[1] http://www.racom.eu/eng/products/m/ripex/h-menu.html#com_par
[2] http://www.racom.eu/eng/products/m/ripex/h-menu.html#slave
[3] http://www.racom.eu/eng/products/m/ripex/h-menu.html#modbus

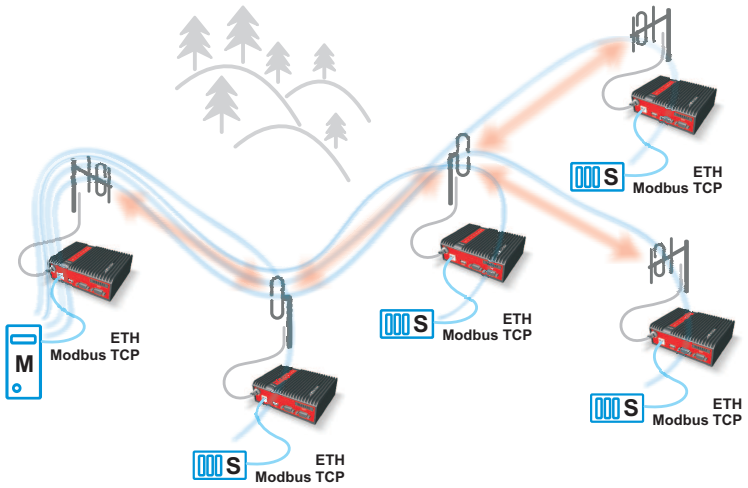


Fig. 8.3: Modbus TCP

8.3. Modbus TCP, local TCP/IP connection

Note - Only works in Router mode.

TCP connection is established only locally between Modbus devices and the connected RipEX units. TCP protocol overhead is not transmitted over the Radio channel. Secured TCP/IP transfer is not necessary because in Router mode every packet in the Radio channel is acknowledged on every radio hop. A packet is therefore repeated directly in the part of the network where it is lost, not across the entire radio network as in TCP/IP. This improves latency and increases network throughput.

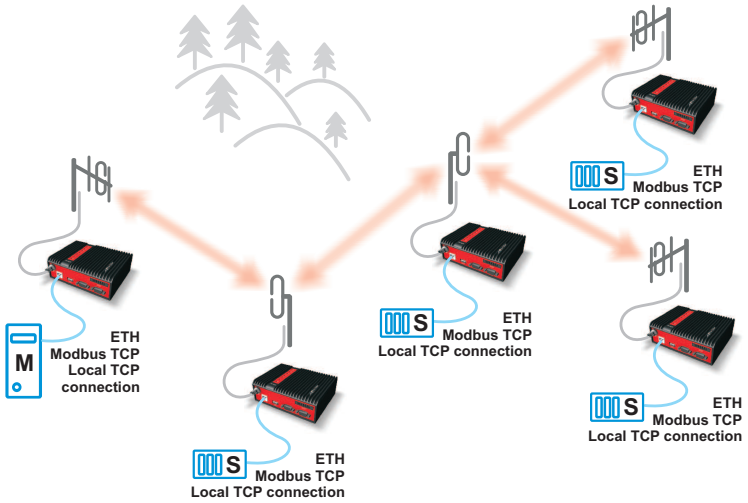


Fig. 8.4: Modbus TCP local

Set your Modbus TCP Master to use a single IP to communicate with Modbus TCP Slaves (RipEX ethernet IP) and set TCP port to 502. Communication begins on port 502 from where it is redirected to other RipEX ports, corresponding to the individual RTU's, based on negotiation with the Modbus TCP Master.

To set up RipEX connected to Modbus TCP Master:

- Set Modbus TCP/RTU to On. Type the port number on which the connected Modbus TCP Master initiates communication, by default 502, into “My TCP Port” field.
- Select how you want to translate Modbus addresses to RipEX IP addresses (using mask or table). Set the UDP interface to Terminal server (TS1-TS5). Set the same TS for remote RipEX’s too.

Note

The maximum number of concurrent TCP/IP connections between a Modbus TCP device and RipEX is set to 10 due to limited computing capacity. (Note: The number of concurrently open TCP/IP connections can be increased using CLI if necessary.) Modbus TCP Master must be set to not open more than 10 TCP/IP connections at any given time.

To set up RipEX connected to Modbus TCP Slave:

- Modbus TCP/RTU - Off
- Terminal Servers - On
- Set the Terminal Server (see RipEX Master settings) to TCP and set My Port to 502. Use the address of the connected Modbus Slave as the destination IP and fill in the destination port number which the connected Modbus Slave device scans for incoming communication.
- Set Protocol to UNI and Mode of Connected device to Slave.

8.4. Master - Modbus TCP, slaves - Modbus RTU

Note - Only works in Router mode.

Master establishes a local TCP connection to RipEX using Modbus TCP protocol, as described in chap. 3. A packet is securely sent over the Radio network to RipEX to which the destination Slave is connected by COM port. The RipEX translates the packet to Modbus RTU format and sends it to the connected Slave using Modbus RTU protocol.

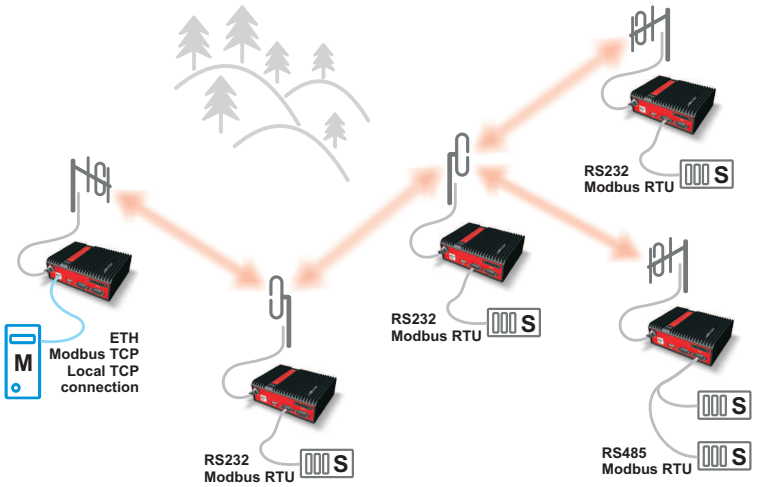


Fig. 8.5: Modbus TCP - RTU

To set up RipEX connected to Modbus TCP Master:

- Select the type of translation from Modbus to RipEX IP address (mask or table), as described in chapter 3.

- Set the UDP interface to COM1 or COM2 depending on the port that the remote RipEX uses to connect to the Slave device.

To set up RipEX connected to Modbus RTU Slave:

- As described in chapter 1 set the appropriate COM to Modbus and the Mode of Connected to Slave.

8.5. Master Modbus TCP, slaves Modbus RTU or Modbus TCP

RipEX radio modems enable full featured cooperation between the Master using Modbus TCP and slave devices using Modbus RTU or Modbus TCP within a single network.

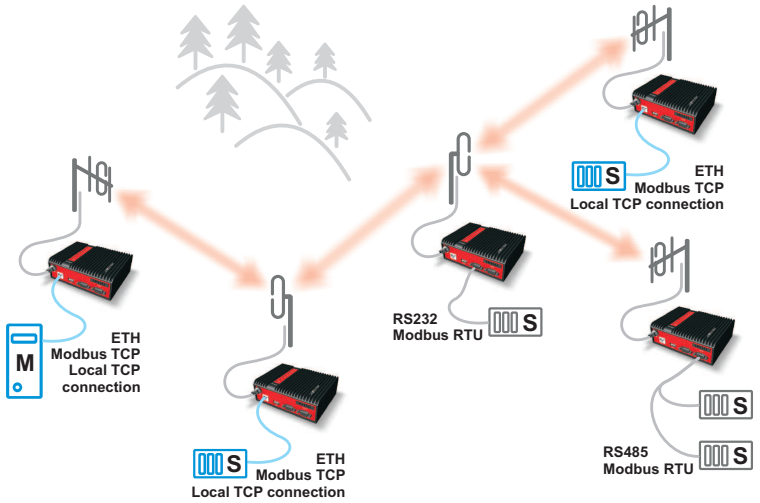


Fig. 8.6: Modbus TCP, Slave RTU or TCP

To set up RipEX connected to Modbus TCP Master:

- Set the translation from Modbus to RipEX IP addresses to table-based, as described in chapter 3.
- For devices connected over Modbus RTU, set the UDP interface to COM1 or COM2 (as in chapter 4).
- For devices connected over Modbus TCP, set the UDP interface to TS1-TS5, as described in chapter 3.
- You can define address ranges in the table for greater ease of use.

To set up RipEX connected to Modbus RTU Slave:

- See chapters 4 and 1 respectively.

To set up RipEX connected to Modbus TCP Slave:

- See chapter 3.

8.6. Multiple Modbus TCP or Modbus RTU Masters and Slaves

Any combination of network designs described in chapters 1–5 is possible. The only limitation is that a Master with Modbus RTU cannot communicate with a Slave using Modbus TCP.

A Slave with Modbus RTU protocol may simultaneously communicate with masters using Modbus TCP and Modbus RTU. The network will deliver responses only to the Master which issued the queries using the appropriate protocol.

The individual settings are described in chapters 1–5.

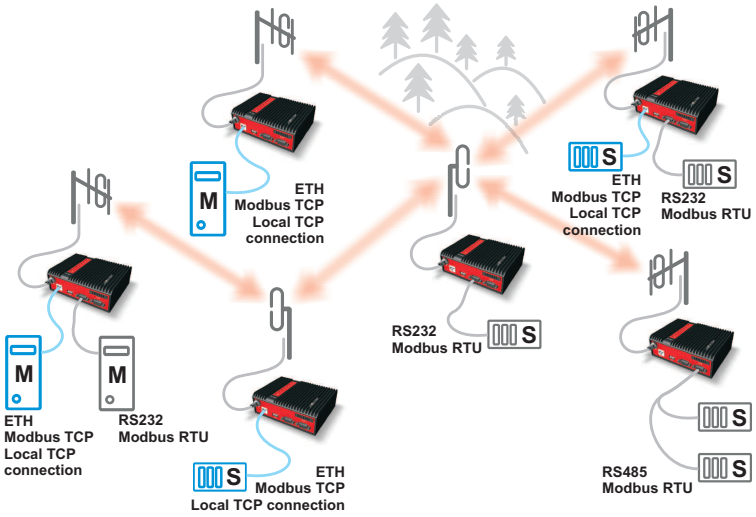


Fig. 8.7: Modbus TCP, Slave RTU or TCP

9. UNI protocol

UNI is the "Universal" protocol utility designed by RACOM. It is not a new SCADA protocol, it can actually process different protocols of different vendors. It supports both the standard MASTER -SLAVE and the MULTI MASTER types of communication. At least one Master is required in the network.

The SCADA protocol to be handled by the UNI has to meet solely the following condition: There has to be an 8 or 16 bit* protocol address in every message generated by a Master station and the address position in all messages has to be the same. The position of address in the reply from an RTU is not relevant, because the reply is always send back to the address where the request originated.

Note

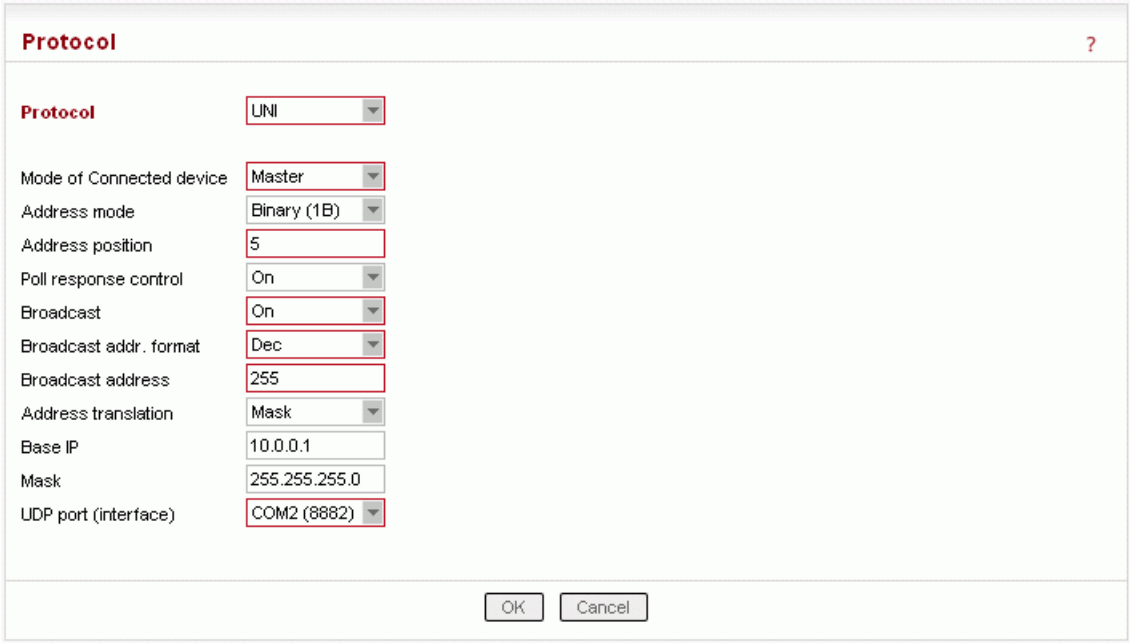
Some SCADA protocols use two byte ASCII address, which is an ASCII representation of an 8 bit address in the hexadecimal format (e.g. "8C" means 8-bit value 0x8C in hex / 140 in decimal notation).

Address bytes for some protocols:

PR2000	3rd Byte
RDS	2nd Byte
Mars-A	8th Byte (without local ACK)
Hirsch	2nd Byte

9.1. MASTER – SLAVE communication

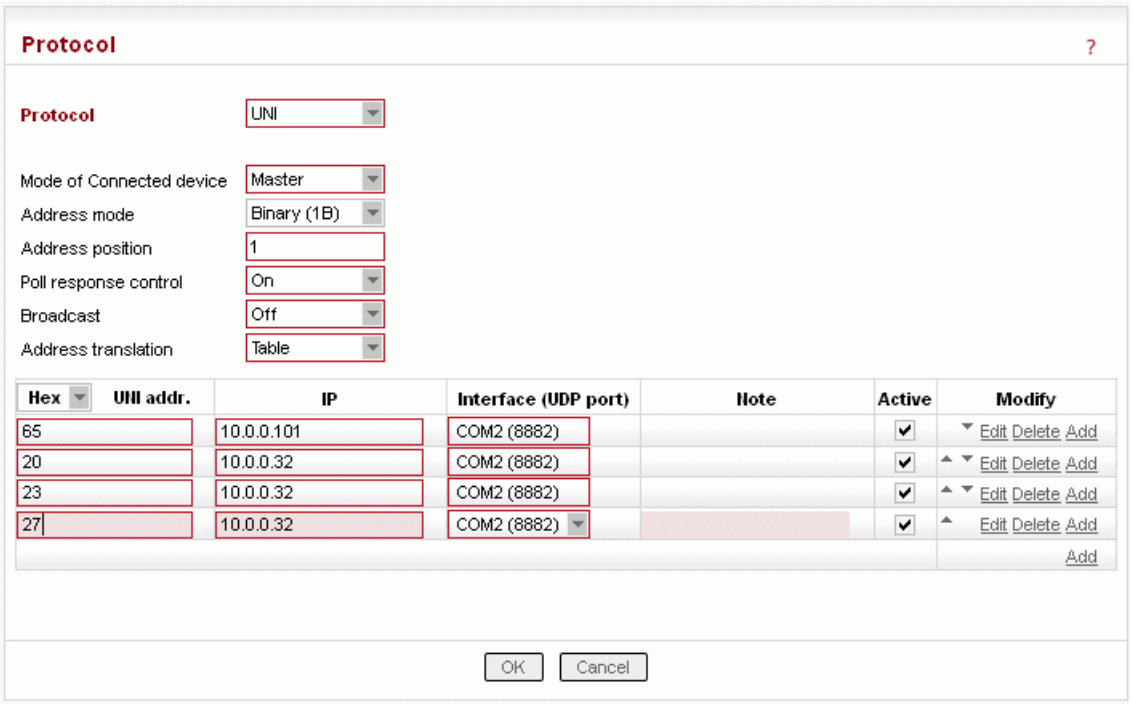
Master reads the address byte defined by configuration and generates the destination IP address using the mask or the translation table. The message is then delivered to that IP address and the respective UDP port (e.g. the port No 8882 which is assigned to the COM2 interface).



An example of Master configuration is in the picture above. The address translation then proceeds as follows:

The 5th byte from the incoming message from SCADA centre is used to replace the last byte of the Base IP and the resulting IP address is used as the destination of the UDP datagram which contains the original SCADA message.
Let assume that the content of 5th byte is 0x65 - then the IP destination address will be 10.0.0.101 and the UDP port 8882.

The translation by a table is more versatile, however it requires an extra line of configuration for every remote in the network. The table has to be used when addresses of RipEX radiomodems and SCADA RTUs do not match or different ports (interfaces) at different remotes have to be configured.



The example of table in the picture above demonstrates a situation when there are three SCADA devices connected to the COM2 of a single RipEX unit over a RS485 bus.

The configuration of a Slave radiomodem is very simple, as demonstrated in the picture below. When a UNI Slave receives the UDP datagram from RF channel, it takes the original SCADA message and transmits it over the respective interface (the COM2 in our example).

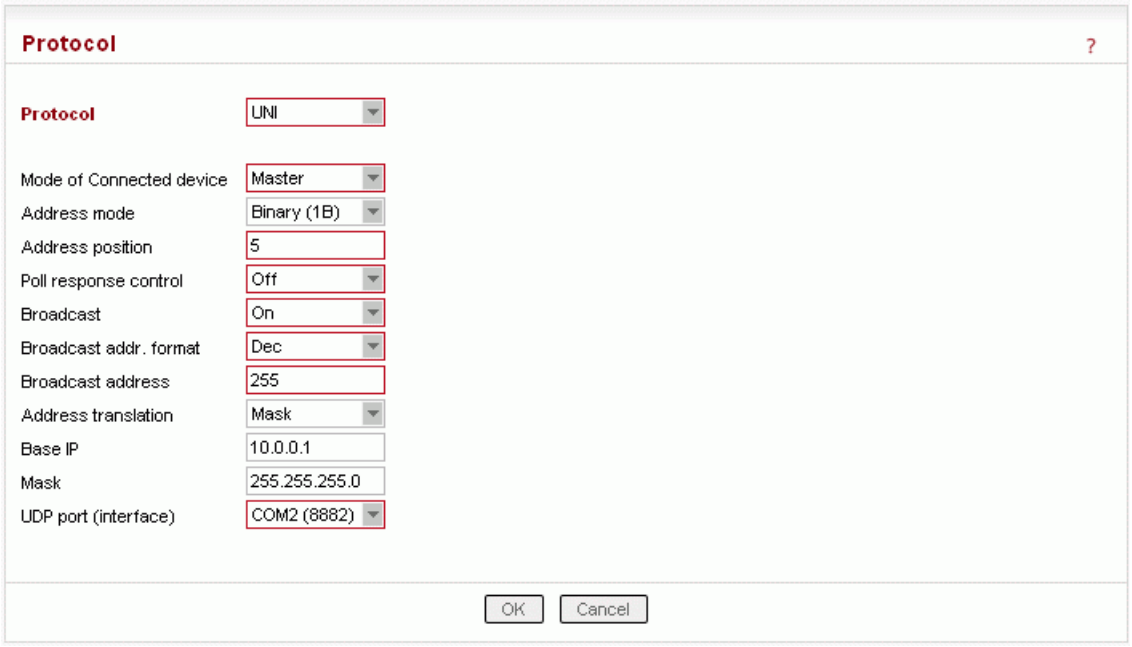
If the SCADA device connected responds to the message within a timeout of 500 ms, the source IP address of the received UDP datagram is used as the destination for the response.(Note only one packet is accepted as a response). When the timeout expires, all messages received by the serial interface are discarded.

9.2. MASTER – SLAVE with several Masters

The behaviour of Master and Slave is exactly the same as in the previous scenario, i.e. a Slave always responds to the address from which the request was sent. If by chance two simultaneous requests from different Masters are received by a slave radiomodem, the RipEX radio modem waits for the first reply from the connected SCADA device before transmitting the request which arrived second. The 500 ms timeout applies again, i.e. when there is no reply for the first request, the second one is transmitted after the timeout expires.

9.3. MASTER – MASTER

The Master - Master communication is possible. The translation of addresses is proceeded with every packet incoming to the RipEX radio modem from connected SCADA equipment, thus it is suitable for SCADA protocols containing the destination address in all packets.



The Poll Response Control has to be set to OFF for the MASTER-MASTER type of communication.

9.4. MASTER UNI – ASYNC LINK SLAVES

The combination of the UNI and the ASYNC LINK protocols is useful for networks where one application master communicates with many slaves and the slaves are allowed to spontaneously send messages to the master. The UNI-Master RipEX§s address is configured as the ASYNC LINK protocol destination address at all the slaves. This arrangement makes the syntax of application protocol messages generated by slave completely arbitrary. All slave messages are transparently delivered to the application master.

Note that, similarly to the MASTER-MASTER mode, the Poll Response Control at the Master RipEX has to be set to Off.

10. Channel access

Method of accessing the radio channel may significantly affect the overall reliability of packet transmission. Even in a simple polling-type application, which never generates more than a single packet at a time, collisions may occur when repeaters are used. The goal of channel access is either to eliminate collisions completely, or to reduce their probability while ensuring that systematic repeated collisions never happen. RipEX provides different channel access methods in different modes and optimum configuration can be found for every communication scheme and network layout.

10.1. Collisions

What is so special about collisions that they deserve that much attention? Well, they are a special case of interference (“friendly fire”, a military reporter would say), which may very seriously harm network performance.

A collision happens, when two (or more) transmissions in the network overlap in time. Radio modem A transmits a packet for B, C transmits for D. In well designed network the respective signal levels (i.e. A received at B, C received at D) do ensure error-less reception. For the period of time when these two transmissions overlap, signal from C at receiver input B and signal A at D act as interference signals, reducing the SNR (Signal to Noise Ratio). If B and D are in the same area, the difference in signal strength is small and so is the resulting SNR at both receivers. Consequently the BER (Bit Error Rate) at both receivers jumps to unacceptable level and none of the packets is successfully received. That is the basic principle of a collision.

There are two very harming features of collisions:

The first is a systematic repeated collision. No application generates a totally random traffic pattern. So it may happen (and it does happen), that a certain sequence of packets in a certain network layout generates a collision and it generates this collision repeatedly, in fact always. The result is that certain specific packets are never delivered, regardless of number of retries set at the application level. Imagine a SCADA system never capable of performing one specific task, while all communication tests report that links are in perfect shape. It would be very tempting to blame the SCADA, while the true problem is a systematic collision, i.e. wrong network design. Ways to avoid such collisions are described further in this document.

The second dangerous feature of collisions is just a direct consequence of probability laws. The most effective communication scheme for many applications is the report-by-exception mode, which can vastly reduce the amount of mainly useless traffic generated by polling-type systems. Report-by-exception means though, that collisions can never be ruled out completely, hence a collision-solving system must be an integral part of the protocol in the radio channel (RipEX in router mode provides such protocol of course). Solving a collision means retransmission, typically a delayed retransmission. Consequently the probability of another packet being generated by the application in the meantime increases by the delay, and it increases at both parties involved in the collision. That results in an increased probability of next collision to happen...and so on. This principle makes report-by-exception networks very sensitive to bursty loads. Whenever the load increases over certain limit (we may say “normal” network capacity), number of collisions grows exponentially, reducing the instant network capacity well below normal situation. Series of lost packets and very long delivery times are the result from the application point of view. While the network for report-by-exception application has to be designed to provide maximum capacity possible, it is recommended to take measures to avoid burst load generation at the application level. Limiting the possible load generated by a single device can help to avoid the whole network collapse just because one remote unit goes suddenly “crazy” (e.g. generates hundreds of “exceptions” per second).

10.2. Bridge mode

In Bridge mode, a packet is transmitted to the radio channel immediately, without any checking whether the radio channel is occupied or not. If other radio was transmitting simultaneously, a collision would occur and both packets would be lost. Consequently Bridge mode can be used only for applications which never generate more than a single message at a time, e.g. master-slave polling applications. Still appropriate measures have to be taken to avoid collisions in special situations.

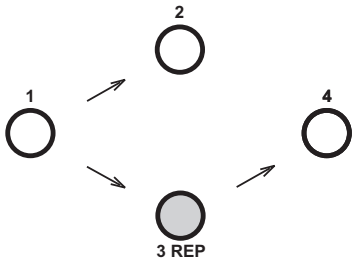
10.2.1. Bridge mode with Repeaters

Repeaters can be used in the Bridge mode in order to extend the radio coverage. Considering the repeated packets, it is necessary to schedule the access to the radio channel to avoid systematic collisions. In a polling-type network, there is a request packet from centre to remote, to which the remote responds immediately. When a remote receives the request directly from the centre, its immediate response would collide with the repeated request, so it would be never received by the centre – a perfect example of a systematic collision.

Packet header contains information about the number of Repeaters on the route, i.e. how many times the packet can be possibly repeated. This number is decremented when passing through each Repeater. The remote radio modem which receives the packet must delay its own transmission for a period. This delay is calculated from the number of the remaining repetitions, the packet length and the modulation rate in the radio channel. Repeaters always transmit immediately, without any delay.

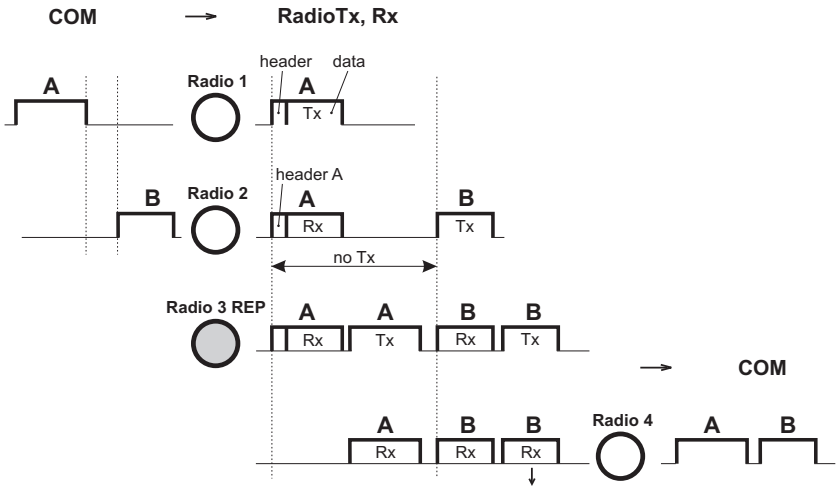
Example:

There are 4 radios in the network operated in the Bridge mode. Everyone can receive each other except Radio 4, which is not able to receive Radio 1 and vice versa. Therefore, in the Radio 3 the Repeater function is turned on, and it mediates the connection between 1 and 4.



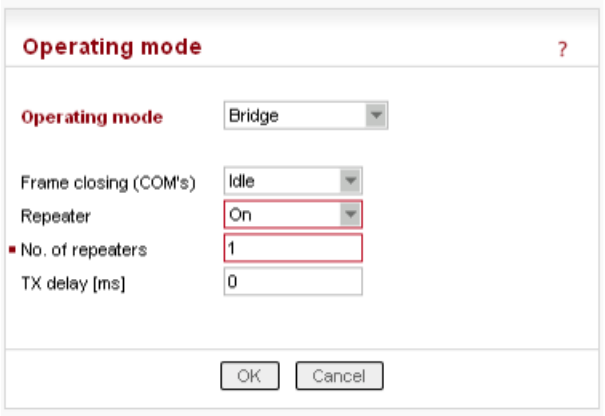
First, packet A is broadcast from Radio 1.

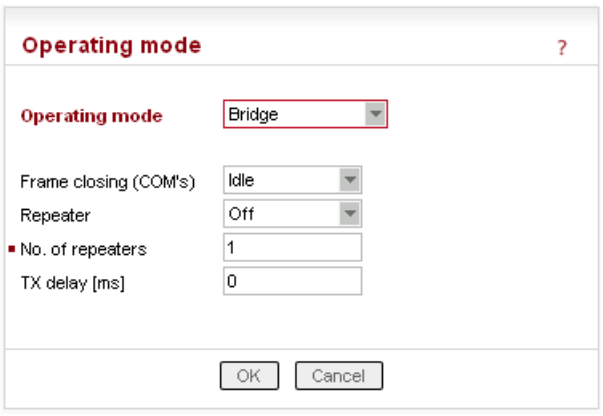
Radio 2 receives Packet A and sends it to its COM. In the instant when it starts the reception of Packet A, Radio 2 calculates (from information in the received packet header and from number of repeaters in its own setting) the time delay which is needed for the delivery of Packet A through the repeater (repeaters). When the response from the connected device arrives via COM (Packet B), the Radio 2 postpones its transmission for the delay.



In the meantime, Radio 3 (Repeater) receives Packet A and repeats it to the radio channel immediately. Radio 4 receives the Packet A and then Packet B and sends them both to the COM. Packet B is also received by Radio 3 and immediately repeated. Whenever a radio receives a copy± of the same packet during the calculated delay, it discards it as a repeated packet. Note that the picture does not show all the packets at all the radios.

Repeater is configured in the Settings / Device / Operating Mode menu, for Radio 3 (left) and Radio 1, 2, 4 (right):





The delay period based on number of repeaters solves the collision between a repeated packet and a possible response. When more than one repeater is used in a Bridge-mode network, collisions between repeated packets from different repeaters may occur. These cannot be solved by simple delays, rather a sophisticated anti-collision protocol is required. The RipEX Router mode is recommended to be used in more complex networks with multiple repeaters. Nevertheless if certain conditions on signal coverage (sometimes non-coverage) among repeaters, centre and remotes are met, the Bridge mode for a polling-type application can be used. See the chapter Bridge mode[1] in RipEX Manual.

[1] http://www.racom.eu/eng/products/m/ripex/ripex-detail.html

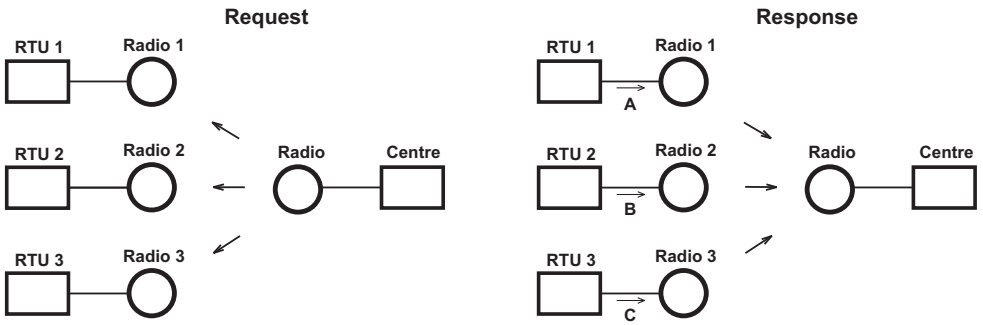
10.2.2. Time division of responses in Bridge mode

There is also the Tx delay setting in the menu. It shall be used in Bridge mode if multiple RTUs connected to slave stations reply to a broadcast query from the centre. It is necessary to spread out their replies to the radio channel in terms of time, otherwise a massive collision occurs. It can be achieved by setting the TX delay parameter to an adequate sequence of TX delays (e.g. 0, 100, 200 ms as in the example below) in individual remote RipEXes. The slave RipEXes will enter the radio channel successively and no collisions will occur.

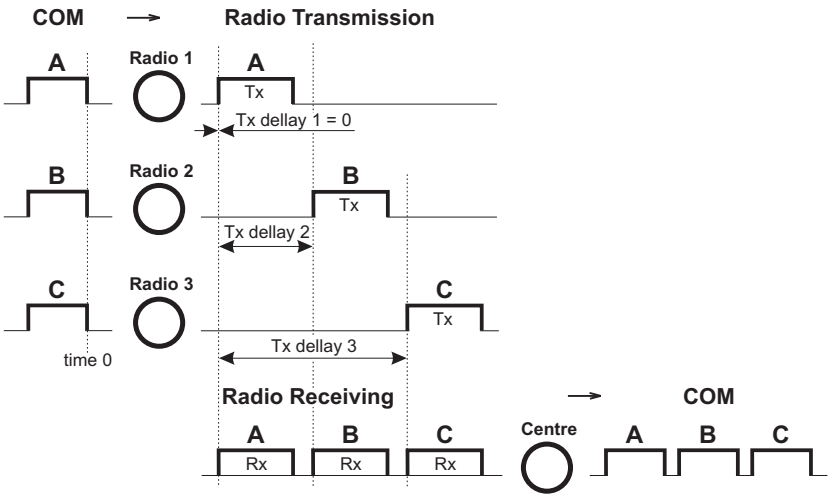
Note: The TX delay applies to every packet that is sent out to the radio channel.

Example:

The Centre broadcasts request and the RTUs 1, 2 and 3 generate the response and send it out to their respective RipEX.



Radios 1, 2 and 3 have the TX delay parameter set to 0, 100 and 200 ms, respectively. Therefore, Radio 1 starts transmitting just after reception of the frame from COM port. Upon 100 ms later, when Radio 1 has completed transmission, Radio2 starts transmitting. Finally, 200 ms after the reception of the packet from RTU, Radio 3 starts its transmission. All three responses are thus sequentially sent to the Centre and no collision happens.



The TX delay parameter coresponds to multiples of maximum packet length expected and shall be set in miliseconds. The packet transmission time through radio channel can be calculated as follows:

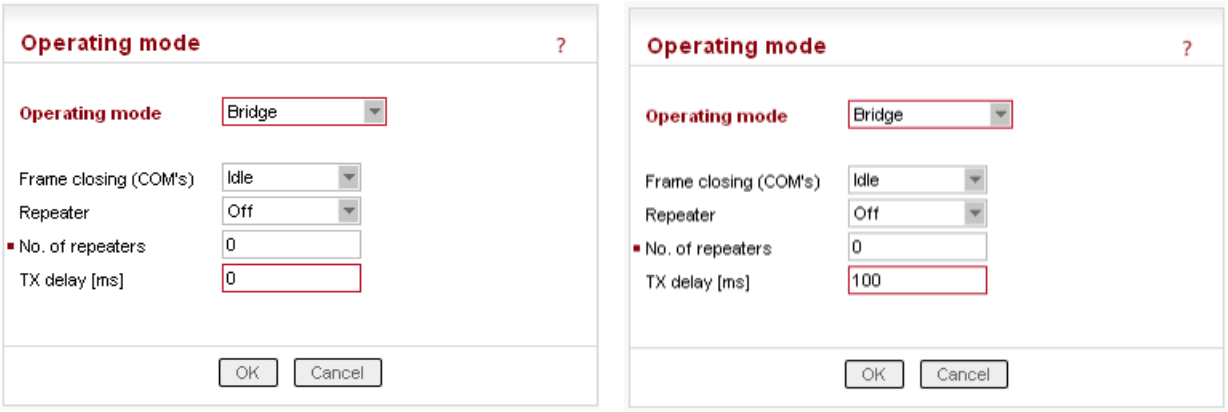
```
t = (n + 12). 8/(b . fec)
```

where:

`t [ms]`	- time needed for the packet transmission
`n [ - ]`	- number of bytes transmitted (consider the longest possible reply from RTU)
`b [kbps]`	- Modulation rate
`fec [ - ]`	- Forward Error Correction
	`fec = 1.00 if FEC = Off`
	`fec = 0.75 if FEC = On`

This calculation gives approximate results (± 3ms). When more accurate calculation is necessary, please check the calculation tool on Racom web pages http://www.racom.eu/eng/products/radio-modem-ripex.html#calculation

TX delay is configured in the Settings / Device / Operating Mode menu, for Radio 1 (left) and Radio 2 (right):

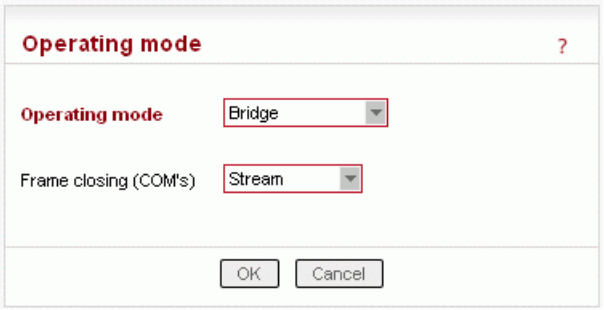


10.3. Bridge mode and COM stream

The COM port in Bridge mode can be switched into the Stream mode. In any other mode, a packet/frame coming to RipEX over any interface has to be received completely before any further processing. In Stream mode the incoming bytes are transmitted to radio channel with minimum possible delay, byte by byte. Consequently nor checks neither processing of the data can be done. All the bytes are simply broadcast to the radio channel and every radio modem which can receive them forwards them immediately to its COM port(s).

Obviously there can not be any repeaters in the Stream mode and no measures against possible collisions can be taken. The responsibility for collision-free communication remains solely with the application. Consequently only simple master-slave polling-type applications, which never respond to broadcasts, can use the Stream mode. This mode should be used solely in applications which would not work when the normal store-and-forward regime is used because of the inevitable delays involved.

The Stream mode is configured in the Settings / Device / Operating Mode menu:



10.4. Router Mode

10.4.1. Channel access in Router mode

The protocol in the radio channel in the Router mode of RipEX uses sophisticated method to prevent and solve collisions. When a data packet with RSS above the configurable threshold or a data packet destined for the RipEX itself are received it leads to the „busy channel“ state, as well as the RipEx's own transmission.

When RipEX evaluates the channel as free, it calculates the Access period – time for which it has to continue monitoring the channel before starting a transmission. Only when the channel stays free for the Access period or more, RipEX starts transmitting whenever a packet destined to radio channel arrives. If channel gets busy, the arriving packets have to wait in a queue and whole process starts from the beginning.

The Access period calculation follows quite complex algorithm, which takes into account RipEX settings, properties of the last packet sent or received and there is – very important – random element. The result is an optimum performance of RipEX's in a report-by-exception network.

10.4.2. Solving collisions in Router mode

When report-by-exception application or multiple-master polling-type one loads the network, collisions can not be avoided completely despite the sophisticated channel access method used. Then a collision-solving algorithm becomes equally important.

The standard protocol feature of sending an Acknowledgement (ACK) to every data packet and retransmitting it when no ACK comes takes care of all possible reasons for packet non-delivery, collisions included. However retransmitting a packet increases the network load and so increases the collision probability. Moreover, it is possible to create a systematic collision by e.g. a regular retransmissions after the initial random collision. Thus the calculation of the retransmission time-out requires a sophisticated approach again. RipEX uses its settings, packet parameters, sequence number of the retransmission and the necessary random element to calculate the time-out.

Retransmission feature is enabled by selecting “On” in the ACK listbox. By deciding on number of Retries you define the very important compromise between the longest possible delivery time and the probability of a packet being lost. Note that this setting does not normally affect the typical (most probable) delivery time in the network, since a typical packet is delivered without retransmissions.

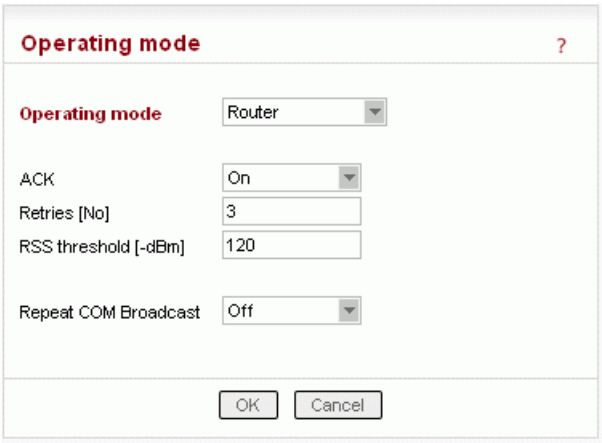
Most applications require their data to be delivered completely and error-free, hence there are message retransmissions at the application level. Note that the RF protocol (i.e. RipEX's) retransmissions are always more effective than the application ones, since the radio modem can use more information from the channel when calculating the retransmission time-out. Moreover, when repeaters are involved, re-

transmitting over a single hop is always faster (and has a greater chance to succeed) than retransmitting over the whole path. Consequently a reasonable approach is to set application time-out to maximum value possible and use an adequate number of Retries in RipEX's in the network. Though the application engineers may find it difficult to understand, such setting will make the application run faster.

There are few exceptions and hitches though. There are applications which rather send a fresh data instead of simply retransmitting the original message. In such case, depending on the frequency of fresh data from the application, the Retries should be set to 1 or ACK switched off completely. Sometimes the application is hard-wired and the retransmission time-out cannot be changed – then it is better to minimize or switch off RipEX's retransmissions again. The trickiest case is when the application centre generates messages to non-existent or switched-off remotes (for any reason). When a remote site is without power (including the RipEX) and the centre continues sending requests to that remote, the last repeater will keep retransmitting these requests for full number of Retries set. More importantly, a long retransmission time-out at the application level is not desirable any more, since it keeps the centre from continuing the polling cycle. Nevertheless in any case it is beneficial to keep the number of application retransmissions at the lowest setting available, i.e. zero if possible, and leave the RipEX network to use the time available for the possible retransmitting.

To calculate the typical and maximum possible delivery time for different settings, please use the calculator on Racom web pages, http://www.racom.eu/eng/products/ripex.html#calculation

The parameters discussed above are configured in the Router operating mode menu. Kindly see the Help pages for further information.



11. ARP Proxy & VLAN

11.1. Introduction

ARP proxy can be used when RTU's IP addresses behind different RipEX units are for any reason within the same IP subnet, typically they do not have routing capabilities.

VLAN feature is typically used when you need to split the network into several logical parts. E.g. to distinguish between management and payload (user data) traffic or among different applications traffic (e.g. various RTU technologies).

Both features can be combined to provide the necessary functionality.

See the following chapters for a detailed description.

11.2. Transparent LAN (ARP Proxy)

Even though RipEX works as a standard IP router, RipEX can interconnect equal IP subnets behind different RipEX units without defining default gateways. It can be done with the **ARP proxy** feature.

Note

See the RipEX manual, Chapter 2.3 Router mode[1] for configuration examples without ARP proxy usage.

RipEX can reply to any ARP request to mimic it has this particular IP address (RipEX can reply to more ARP requests). This feature is typically used when RTU's IP addresses behind different RipEX units are within the same IP subnet and the RTUs do not provide routing capabilities.

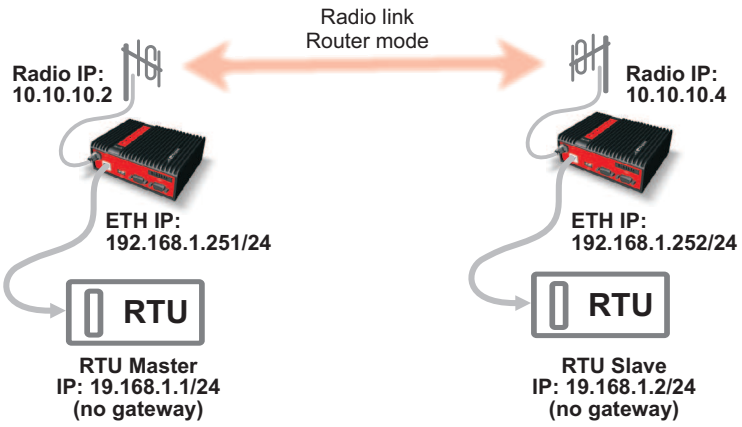


Fig. 11.1: Basic ARP proxy usage

In this diagram RTUs do not have routing capabilities (i.e. RTU expects its counterpart is within the same physical Ethernet LAN). If the RTU Master starts to communicate with RTU Slave, it requests the RTU Slave's MAC address. The RTU Slave is a member of the same physical LAN so the RTU Slave does not reply. However, when RipEX (radio IP 10.10.10.2) has ARP proxy enabled, it replies to this ARP request.

[1] http://www.racom.eu/eng/products/m/ripex/ripex-detail.html#router_mode

So with the ARP proxy functionality, local RipEX can mimic any IP address and reply to ARP requests. In our case, the RTU Master would consider the RipEX MAC address as the Slave MAC address. And with the appropriate routing rules in RipEX units, we can achieve the needed interconnectivity. We do not need to set anything on the connected RTUs – no gateway, no routing rules.

Important

Be very careful when using this feature, ARP proxy can disable all the traffic on the LAN!

Note

- You can combine the ARP proxy feature with a TCP proxy and Terminal Servers. See the respective help in the RipEX web interface for details.
- RipEX does not transmit broadcast packets via the radio link with the ARP proxy feature.

11.3. Transparent VLAN

The VLAN tag (802.1Q protocol) is a 4B field in the Ethernet frame. It is inserted between the MAC address and EtherType/Length fields of the original frame.

The VLAN packet is defined by two main parameters:

VLAN tag	– VLAN Identifier (VID) is also called “VLAN number”. It is 12 bits long so we can have up to 4096 VLANs (0x0000 and 0xFFF values are reserved).
Priority Code Point (PCP)	– a 3bit field which refers to the IEEE 802.1p priority. It indicates the frame priority level. Possible values are from 0 (best effort) to 7 (highest priority); 1 represents the lowest priority. These values can be used to prioritize different traffic classes (voice, data, …).

See the following example:

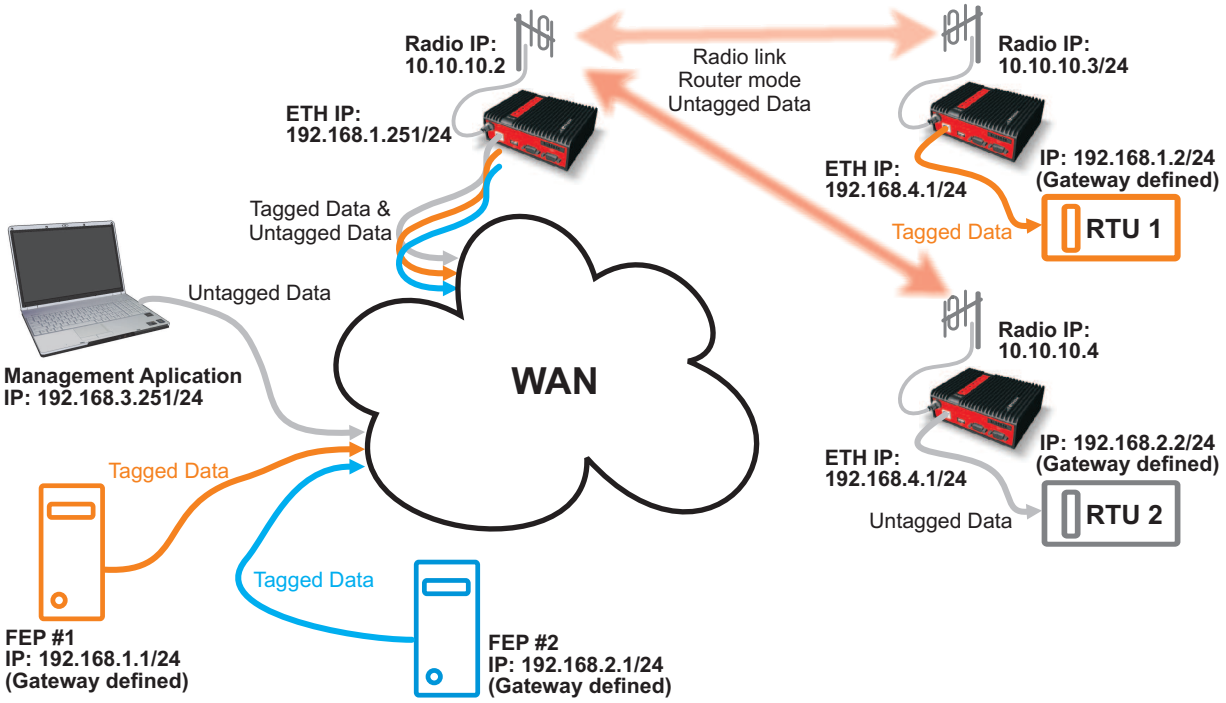


Fig. 11.2: VLAN diagram

As you can see in Fig. 11.1, “Basic ARP proxy usage”, we have individual VLANs for Management and two distinct technologies, each with its own IP subnet.

Note

You can combine the VLAN feature with a TCP proxy and Terminal Servers. See the respective help in the RipEX web interface for details.

11.4. Configuration Examples

In this chapter, we will go through several examples in order to explain ARP proxy and VLAN features in practice. All examples will have the same hardware configuration and we will alter the software settings only (ARP proxy, VLAN tagging, routing, …). Regular PCs will be used instead of RTUs.

Please follow the examples one by one to fully understand the configuration differences and benefits of various solutions.

11.4.1. No ARP Proxy and No VLAN

We will begin with a basic configuration example without using ARP proxy or VLANs. See the following diagram:

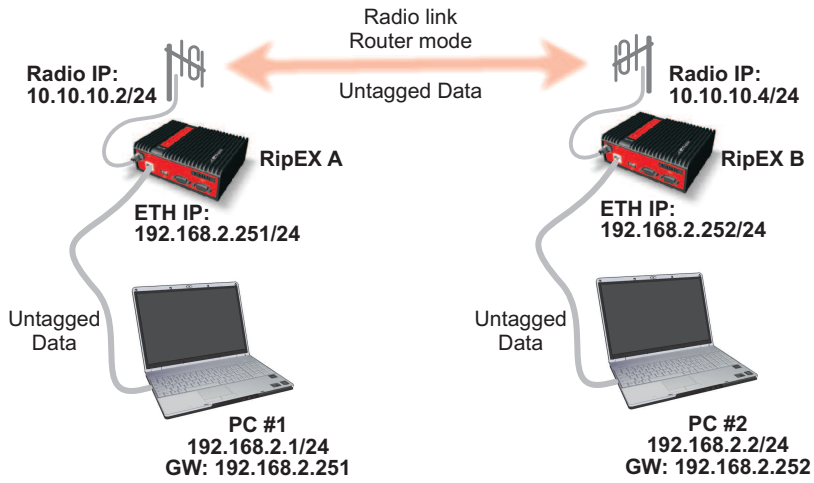


Fig. 11.3: Basic configuration diagram

This example does not reflect the common configuration, because the computers share the same IP subnet, but behind different RipEX units in the Router mode. Usually the RipEX units would connect different IP subnets. This can easily be done with ARP proxy, but in this example, we can configure it with special routing rules.

Note

Do not connect the PCs via X5 converter, but use the Ethernet interface. You can use the X5 converter just for configuration steps, not the connectivity tests.

RipEX Configuration

To access the first RipEX unit, go to the Settings and name it RipEX A. Set the following IP addresses:

- Radio IP address: 10.10.10.2, mask 255.255.255.0
- Ethernet IP address: 192.168.2.251, mask 255.255.255.0

On the second unit, set the name to RipEX B and configure it with the appropriate IP addresses:

- Radio IP address: 10.10.10.4, mask 255.255.255.0
- Ethernet IP address: 192.168.2.252, mask 255.255.255.0

See the RipEX A settings on the following screen-shot.

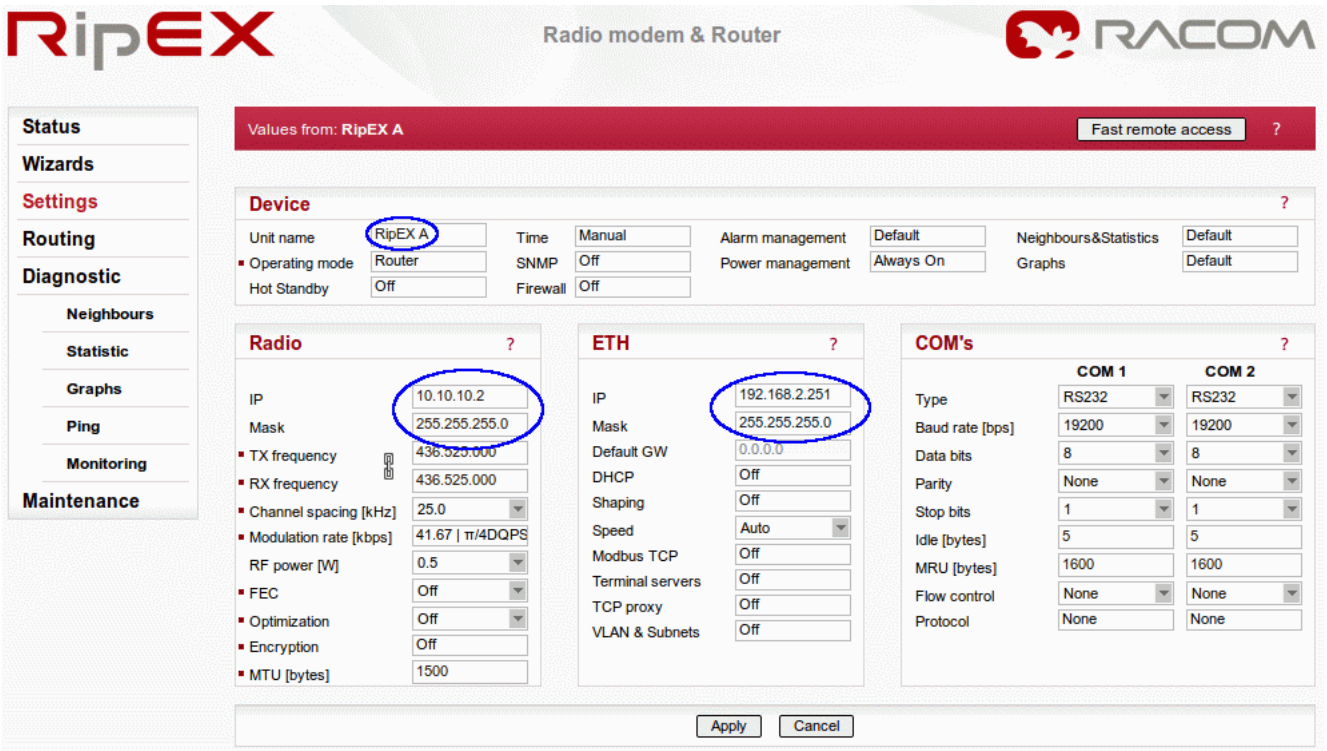


Fig. 11.4: RipEX A settings

Do not forget to set the same TX/RX frequencies, Channel spacing, Modulation rate and other parameters on both RipEX units. **Do not** enable ARP proxy or VLAN.

The next step is to set Routing (see the **Routing** menu). Configure RipEX A with these routing rules:

- Destination: 192.168.2.252/32, Mask: 255.255.255.255, Gateway 10.10.10.4
- Destination: 192.168.2.2/32, Mask: 255.255.255.255, Gateway 10.10.10.4

RipEX B will have very similar routes:

- Destination: 192.168.2.251/32, Mask: 255.255.255.255, Gateway 10.10.10.2
- Destination: 192.168.2.1/32, Mask: 255.255.255.255, Gateway 10.10.10.2

Do not forget to activate both routes. You can also add a note to each route. See the RipEX A Routing example:

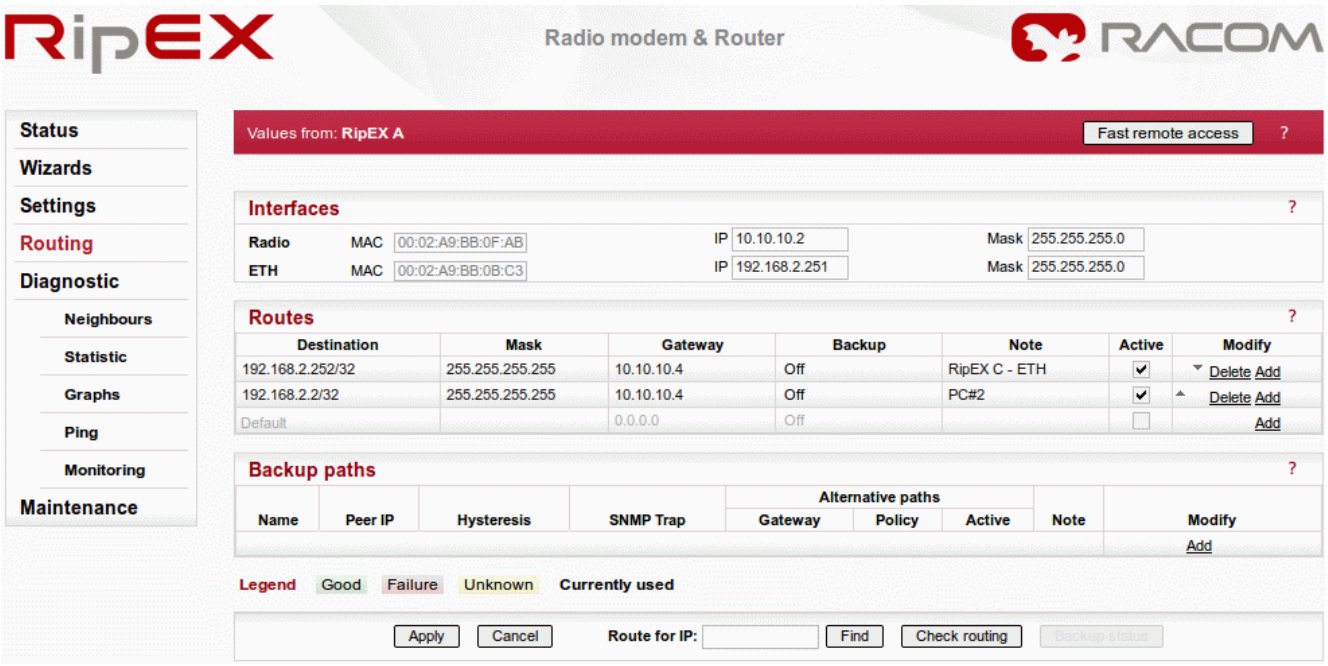


Fig. 11.5: RipEX A Routing

Computer Configuration

When we have successfully configured both RipEX units, we can proceed with computers settings.

- **PC #1:** IP address: 192.168.2.1, Mask: 255.255.255.0, Default Gateway: 192.168.2.251
- **PC #2:** IP address: 192.168.2.2, Mask: 255.255.255.0, Default Gateway: 192.168.2.252

Note

If you do not know how to configure these computers, see the RipEX manual, http://www.racom.eu/eng/products/m/ripex/bench-test.html#connect-PC.

In the common configuration with two different IP subnets behind our RipEX units, we would not need any further action to get the end-point connectivity. In this example, we must add two routes on both computers.

To add routing rules in Windows, you need to execute **Windows Command Processor** (cmd). Click on the **Start** button and then type *Command Prompt* or *cmd* in the **Start Search** field. Select the Command Prompt icon.

After the Command Prompt window appears, type the following commands on PC #1:

- `route add 192.168.2.252 mask 255.255.255.255 192.168.2.251`
- `route add 192.168.2.2 mask 255.255.255.255 192.168.2.251`

You also need to add similar routing rules on PC #2:

- `route add 192.168.2.251 mask 255.255.255.255 192.168.2.252`
- `route add 192.168.2.1 mask 255.255.255.255 192.168.2.252`

Note

You need Admin privileges to add a route in Windows 7.

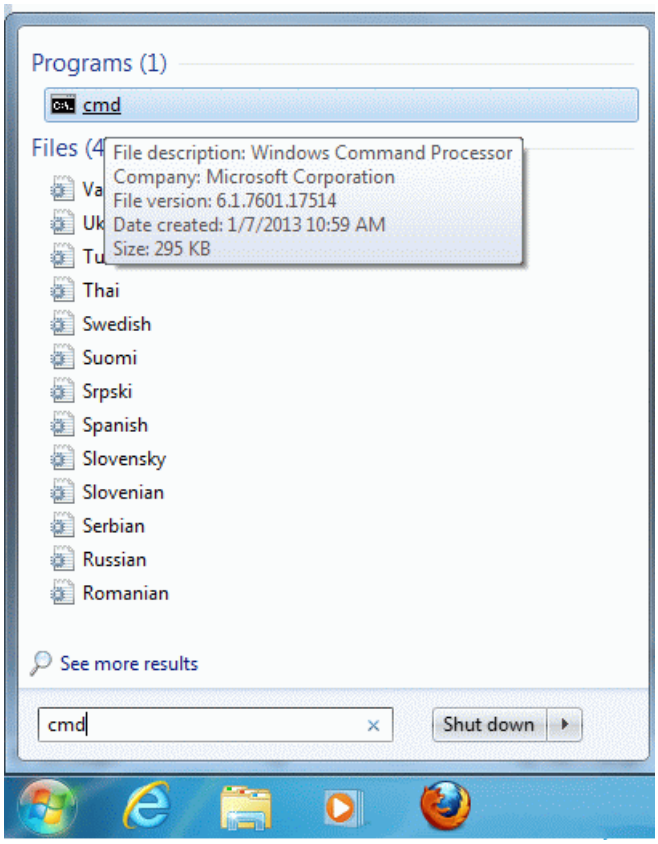


Fig. 11.6: Command Prompt

Test the Connectivity

Check the connectivity by executing a **ping** command, which is also executed from the *Command prompt*. Type "*ping 192.168.2.1*" or "*ping 192.168.2.251*" if you are executing the ping from the PC #1 and check the results. You can also try the other direction, just switch IP addresses. See the following example:

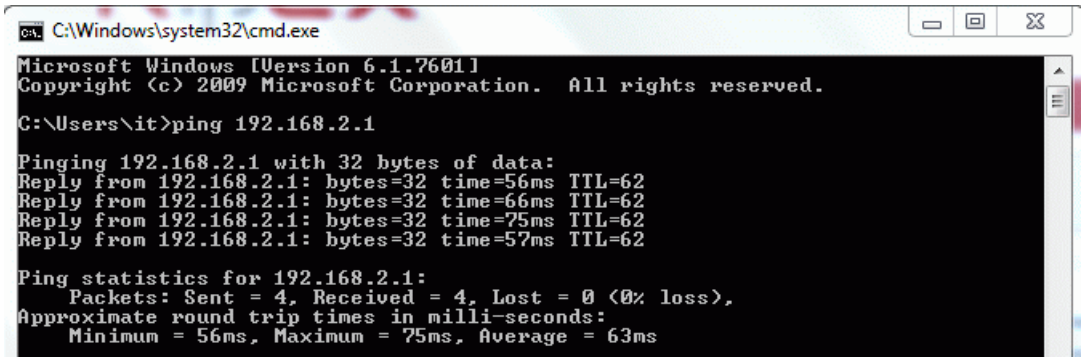

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\it>ping 192.168.2.1
Pinging 192.168.2.1 with 32 bytes of data:
Reply from 192.168.2.1: bytes=32 time=56ms TTL=62
Reply from 192.168.2.1: bytes=32 time=66ms TTL=62
Reply from 192.168.2.1: bytes=32 time=75ms TTL=62
Reply from 192.168.2.1: bytes=32 time=57ms TTL=62
Ping statistics for 192.168.2.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 56ms, Maximum = 75ms, Average = 63ms

Fig. 11.7: Ping results (Basic configuration)

Note

If the ping is not successful, try to turn the Windows firewall off. It can block the ping packets.

11.4.2. ARP Proxy

If we would not have computers as the end-stations, but only simple RTUs, it may happen that routes and default gateways cannot be configured. In this case, we need to reach the connectivity via the ARP proxy feature. See the diagram:

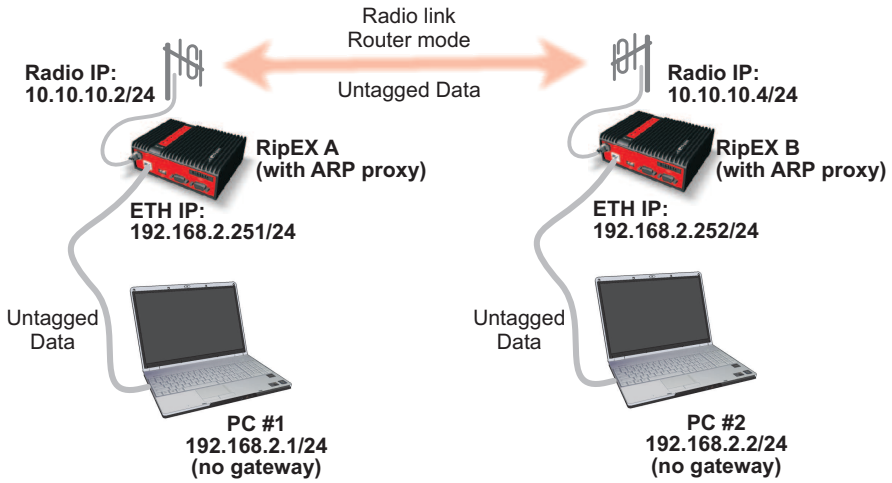


Fig. 11.8: ARP proxy configuration diagram

RipEX Configuration

On both RipEX units we have almost everything already configured. Just go to the **Settings** menu and click on the **VLAN & Subnets** button.

Turn the feature on, and check the ARP proxy option on both units. Confirm and apply the changes.

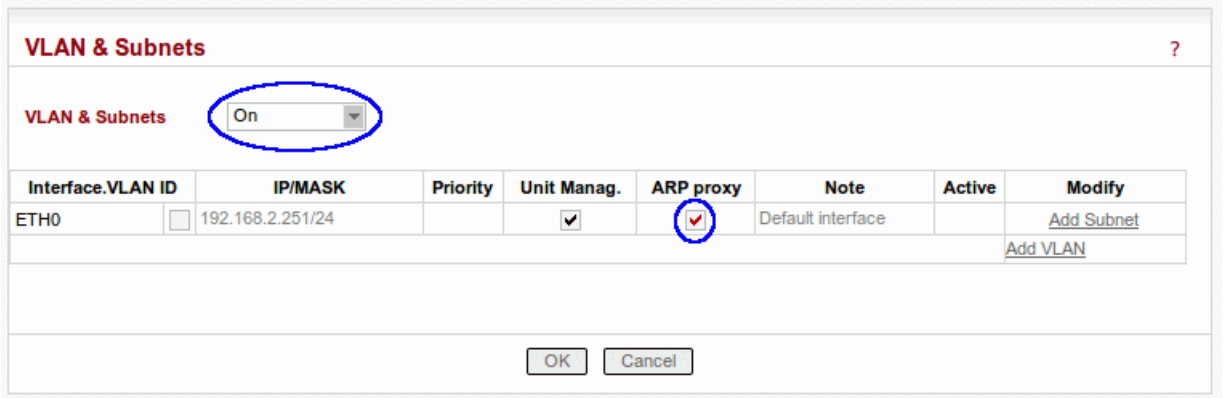


Fig. 11.9: Enabling the ARP proxy

You do not need to change the routing rules. Just remember that the ARP proxy feature works for all destination IP addresses in the RipEX routing table. RipEX will not mimic ARP proxy replies to any other IP address.

Add routing rules to enable ARP proxy on other IP addresses (e.g. if wanting to use the ARP proxy for IP addresses 192.168.2.8-15, add the IP subnet 192.168.2.8/29 into the routing rules).

Computer Configuration

Both computers have the same IP addresses as in the basic configuration example. Just remove the default gateway.

- **PC #1:** IP address: 192.168.2.1, Mask: 255.255.255.0
- **PC #2:** IP address: 192.168.2.2, Mask: 255.255.255.0

You need to delete the routing rules we added previously, just go the the Command prompt again and type in the following commands:

- **PC #1:**
  - route delete 192.168.2.252 mask 255.255.255.255 192.168.2.251
  - route delete 192.168.2.2 mask 255.255.255.255 192.168.2.251
- **PC #2:**
  - route delete 192.168.2.251 mask 255.255.255.255 192.168.2.252
  - route delete 192.168.2.1 mask 255.255.255.255 192.168.2.252

Test the Connectivity

The test is exactly the same as described in Chapter the section called “Test the Connectivity”.

The most important thing to remember with the ARP proxy example is that we did not need to configure any default gateway or routing rules on the computers (RTUs). Thanks to this, we can even add “simple” RTUs to our network and we can have the same IP subnets behind different RipEX units.

Tip

Give careful thought to the network design, because a good design can dramatically reduce the number of necessary routing rules in the RipEX routing table.

Example 11.1. Routing rules

You have four end stations with IP addresses 192.168.2.1, .2.2, .2.5 and 2.6 and you need two of them behind RipEX A and two of them behind RipEX B. With 192.168.2.1 and .2.2 behind RipEX A, you will need to add only one rule in the RipEX B: 192.168.2.4/30 via RipEX A. Otherwise you will need to add two rules (e.g. with .2.1 and .2.5 IP addresses).

11.4.3. VLAN

We will explain two similar examples to show the VLAN functionality.

VLAN on “One End”

In this example, we will have a VLAN ID 2 used between RipEX A and PC #1. RipEX management traffic on the same Ethernet port would be untagged.

Traffic on the radio channel is always untagged.

Traffic between RipEX B and PC #2 will be also untagged.

See the following diagram:

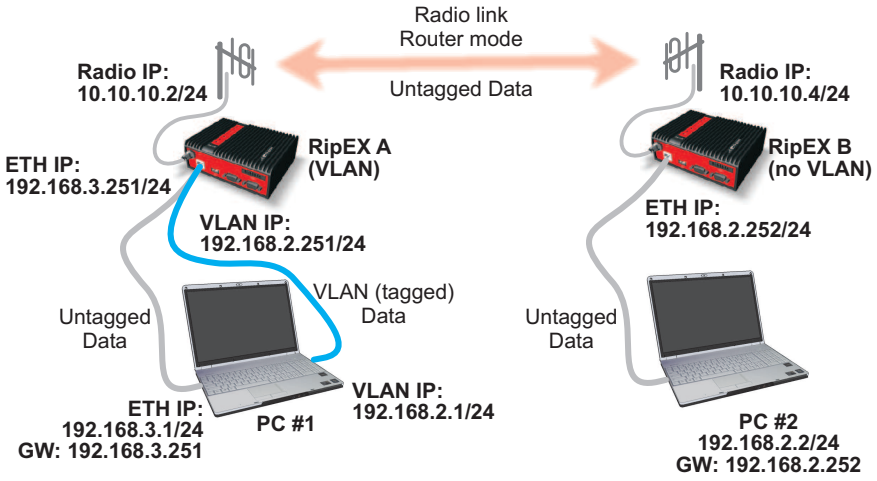


Fig. 11.10: VLAN configuration diagram

RipEX Configuration

The configuration on RipEX A will be a little more complicated. There will be two subnets, one for VLAN and one for other traffic. Go to the **Settings** menu and change the Ethernet IP address to 192.168.3.251. Then click on the **VLAN & Subnets** button and add a new VLAN – we will use VLAN ID 2 with an IP address 192.168.2.251.

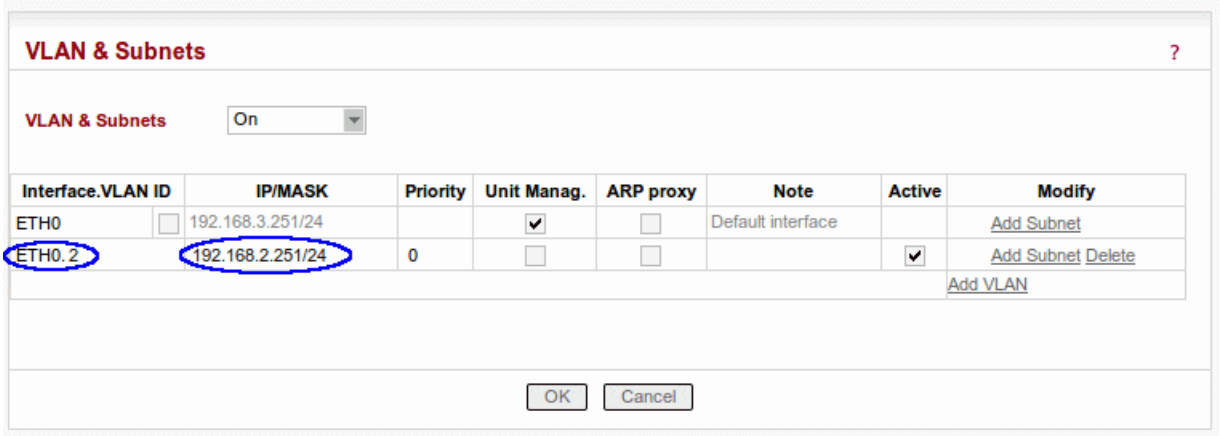


Fig. 11.11: RipEX A – VLAN configuration

On RipEX B, turn the VLAN & Subnets option off.

The routing rules can stay exactly the same as in the previous ARP proxy example on both RipEX units. If you want to have RipEX A management (ETH) IP subnet reachable from RipEX B, you can add this routing rule: 192.168.3.0/24 via 10.10.10.2. But this is not necessary for the end-station connectivity.

Computer Configuration

PC #2 IP configuration is the same:

- IP address: 192.168.2.2

- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.2.252

Setting of PC#1 is not so straightforward. Please set the following parameters:

- IP address: 192.168.3.1
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.3.251

As you can see we are connected to RipEX A within the 192.168.3.0/24 management IP subnet. But we still need to configure the VLAN interface. This step depends very much on the Operating system (OS) you use. We will describe the necessary steps in Ubuntu 12.04 and will give you a short Windows 7 example too.

Ubuntu 12.04

In the command prompt, run the following commands:

- **modprobe 8021q**
- **vconfig add eth0 2**
- **ip link set eth0.2 up**
- **ip link set mtu 1496 dev eth0.2**
- **ip addr add 192.168.2.1/24 dev eth0.2**

The most important command is **vconfig**, which creates the VLAN interface called eth0.2. We enabled the interface, decreased the MTU because 4 additional bytes are added to each frame due to the VLAN tag and of course we assigned the IP address to the interface.

The last two commands create routes so any packet destined to the 192.168.2.2 or 192.168.2.252 is routed via 192.168.2.251 gateway (RipEX VLAN interface).

Windows 7

There is no tool like vconfig in Windows 7. The VLAN features depend on the network adapter and driver installed. Please see the respective download sites of your network card to obtain the correct VLAN enabled driver.

Note

There is also the possibility that your network card will not support VLANs at all.

To see what network card and driver you have, go to **START** → **Control Panel** → **System and Security** → **Device manager** → **Network Adapters** menu. Here you should see your network card. Right click on it and select the Properties option.

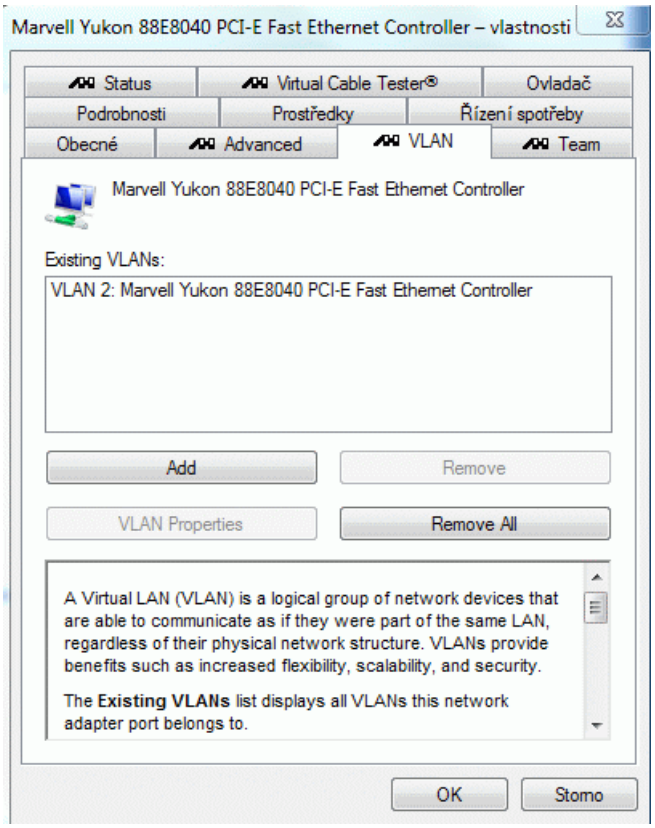

Marvell Yukon 88E8040 PCI-E Fast Ethernet Controller – vlastnosti
Status
Virtual Cable Tester®
Ovladač
Podrobnosti
Prostředky
Řízení spotřeby
Obecné
Advanced
VLAN
Team
Marvell Yukon 88E8040 PCI-E Fast Ethernet Controller
Existing VLANs:
VLAN 2: Marvell Yukon 88E8040 PCI-E Fast Ethernet Controller
Add
Remove
VLAN Properties
Remove All
A Virtual LAN (VLAN) is a logical group of network devices that are able to communicate as if they were part of the same LAN, regardless of their physical network structure. VLANs provide benefits such as increased flexibility, scalability, and security.
The Existing VLANs list displays all VLANs this network adapter port belongs to.
OK
Storno

Fig. 11.12: Adding VLANs in Windows 7

On the example, we added a VLAN 2 interface. See the respective network card manuals for more details.

If you were successful in adding a new VLAN interface. You should see this interface among other physical network interfaces. Set the IP address, mask and gateway as usual:

- IP address: 192.168.2.1
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.2.251

Now you just need to add routes to the 192.168.2.2 and 192.168.2.252 IP addresses. Run the Command prompt and type:

- **route add 192.168.2.252 mask 255.255.255.255 192.168.2.251**
- **route add 192.168.2.2 mask 255.255.255.255 192.168.2.251**

Note

You need Admin privileges to add a route in Windows 7.

Test the Connectivity

The test is exactly the same as described in prvious chapters.

You can run the Monitoring feature in RipEX to capture packets on the radio/Ethernet interfaces and see Ethernet VLAN tags and other valuable information. See the following example:

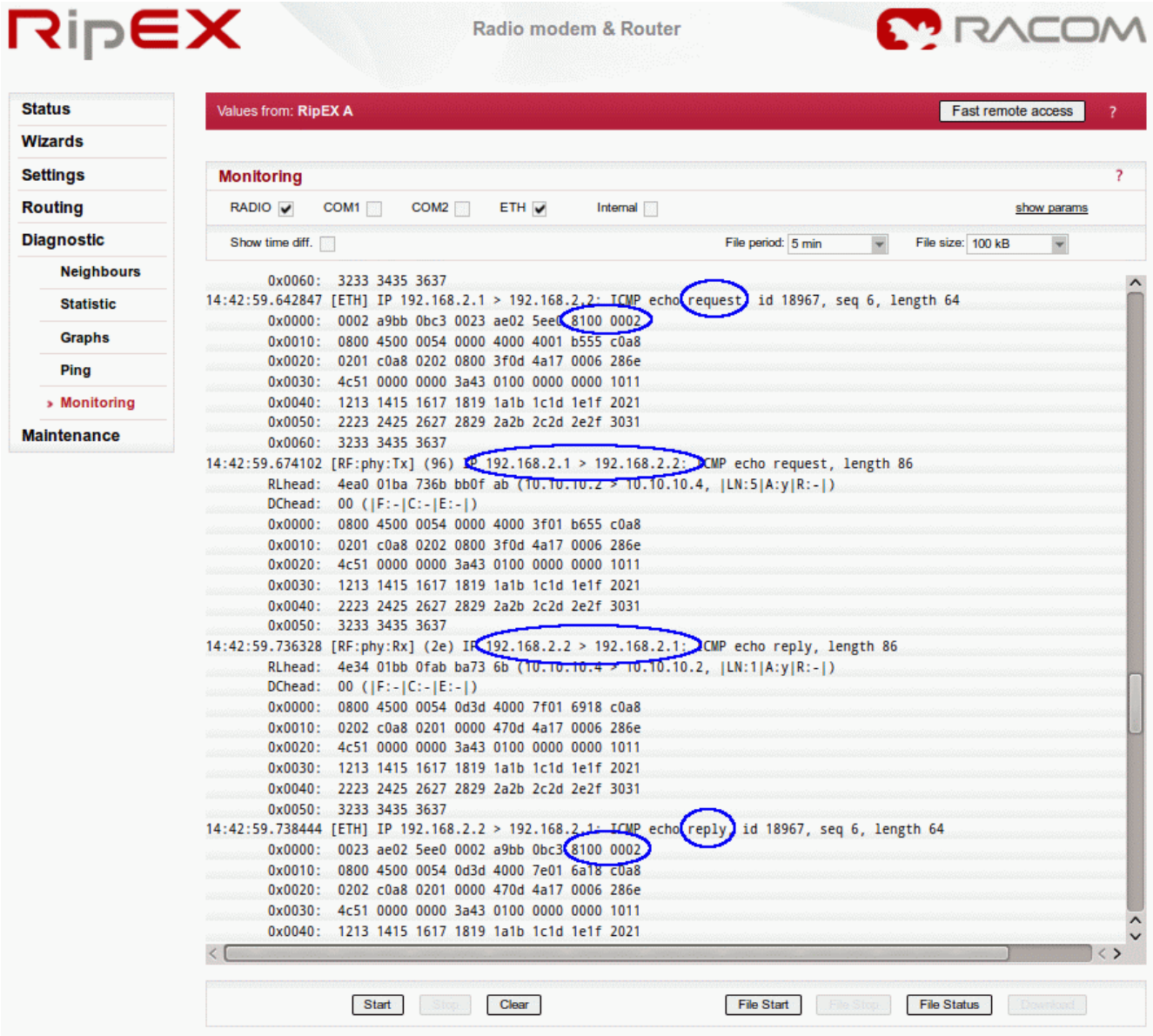


Fig. 11.13: Monitoring ping packets with VLAN tags

VLAN on “Both Ends”

We can also configure VLANs on both RipEX units so the VLAN (tagged) data will be transmitted via the Ethernet link between PC #2 and RipEX B too. However, traffic is always untagged on the radio channel.

See the following diagram:

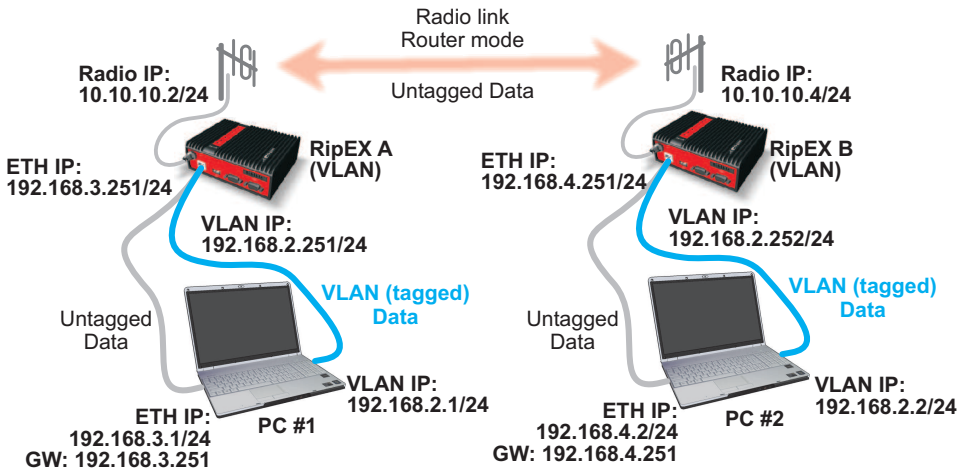


Fig. 11.14: VLAN configuration diagram #2

RipEX Configuration

RipEX A has the same configuration as in the previous example. If you want to test the connectivity of RipEX ETH interfaces, you need to add this routing rule:

- Destination: 192.168.4.0/24, Mask: 255.255.255.0, Gateway 10.10.10.4

RipEX B needs several changes. Change the Ethernet IP address to 192.168.4.252 with the mask 255.255.255.0. Now go to the **VLAN & Subnets** menu, enable the feature and add a new VLAN – we will use VLAN ID 2 with the IP address 192.168.2.252.

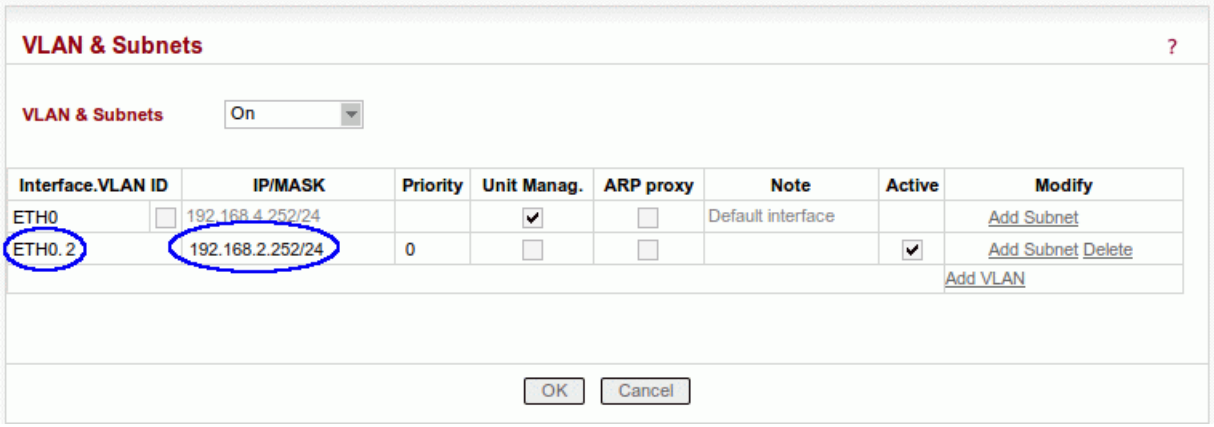


Fig. 11.15: RipEX B VLAN configuration

The VLAN ID is the same as used on RipEX A, but we can set any ID when needed.

Note

You can try to enable VLAN on the default interface after you complete this example.

The RipEX B routing table consists of three rules:

- Destination: 192.168.2.251/32, Mask: 255.255.255.255, Gateway 10.10.10.2
- Destination: 192.168.2.1/32, Mask: 255.255.255.255, Gateway 10.10.10.2
- Destination: 192.168.3.0/24, Mask: 255.255.255.0, Gateway 10.10.10.2

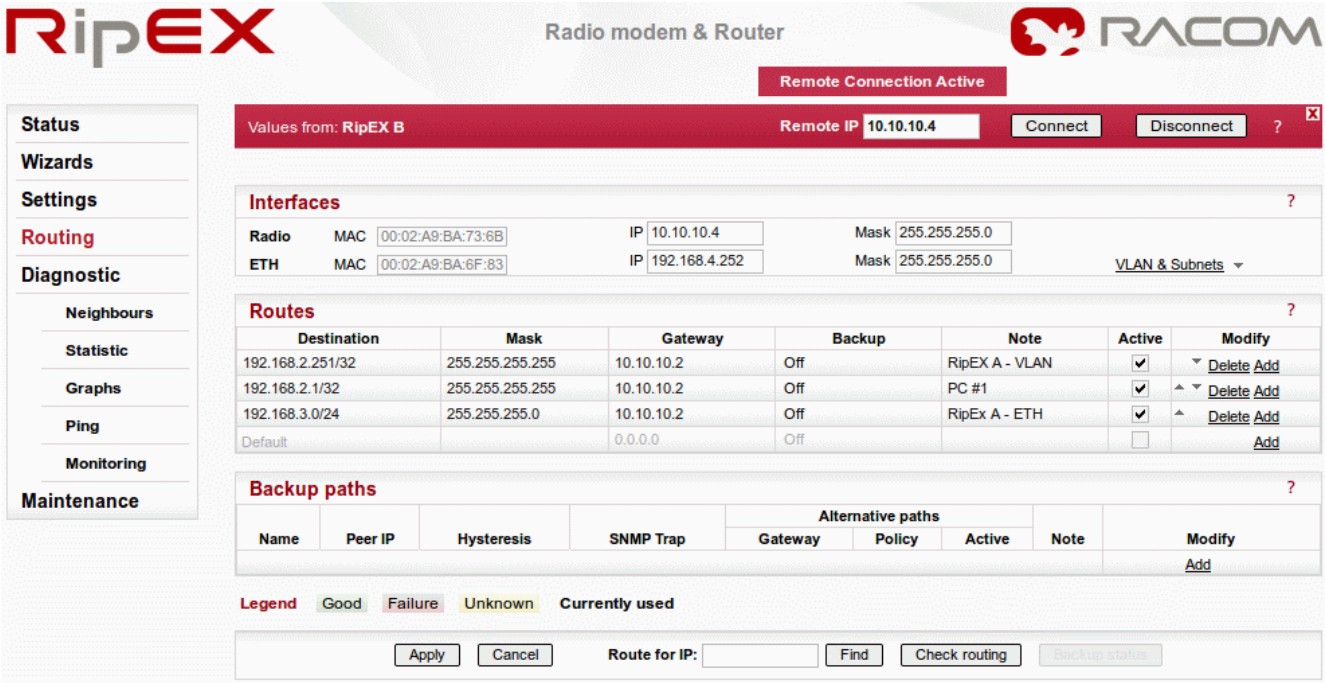


Fig. 11.16: RipEX B Routing table

Computer Configuration

We do not need to change anything on PC #1. PC #2 needs the following changes:

- IP address: 192.168.4.2, mask 255.255.255.0, gateway 192.168.4.252

Now we need to add the VLAN interface with an ID 2. See the procedure in the previous example.

When you have added the VLAN interface, add the following routing rules:

- **route add 192.168.2.251 mask 255.255.255.255 192.168.2.252**
- **route add 192.168.2.1 mask 255.255.255.255 192.168.2.252**

Note

You need the admin privileges to add a route in Windows 7.

Test the Connectivity

Follow the steps described in any of previous chapters called “Test the Connectivity”. You should be able to ping any VLAN or Ethernet IP address from any unit.

Management VLAN

Now you should be experienced enough for the next test. Set another VLAN ID on both computers. Use the same VLAN ID on ETH.0 interface for the **RipEX management**. You will have a “VLAN only” network.

See one of the possible examples:

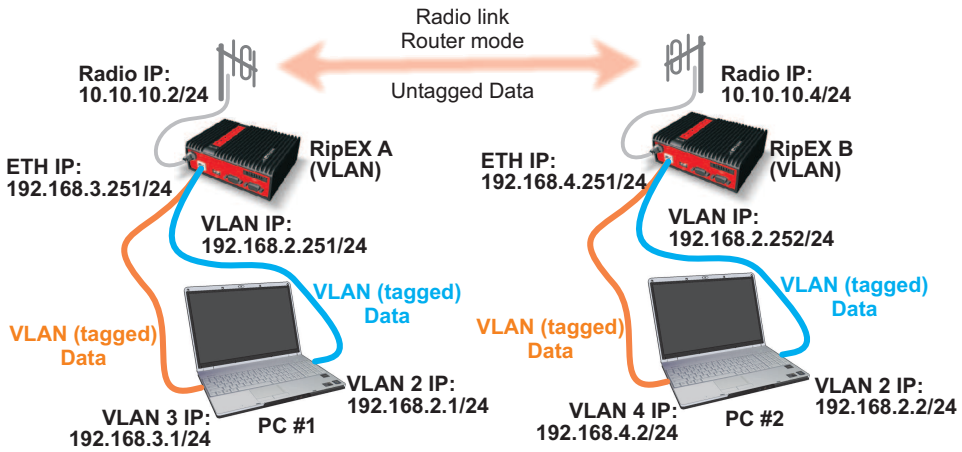


Fig. 11.17: 15 Management VLAN diagram

Note

VLAN 2 is on the same subnet 192.168.2.0/24. VLAN 3 is on the subnet 192.168.3.0/24 and VLAN 4 is on the 192.168.4.0/24 subnet.

11.5. Summary

We have described just a few basic examples of VLAN & ARP proxy usage. Feel free to download the RipEX User manual from http://www.racom.eu/download/hw/ripex/free/eng/ripex-m-en.pdf or the Application notes from http://www.racom.eu/download/hw/ripex/free/eng/ripex-app-en.pdf to conduct further tests.

Do not hesitate to contact us if you have any questions:

RACOM technical support team
E-mail: <support@racom.eu>
Tel.: +420 565 659 511

12. Backup routes

12.1. Introduction

RipEX provides **Backup routes** functionality to increase reachability in networks through redundant paths.

See the following example, where we have three possible paths between RipEX A and RipEX B. The direct radio link is set as the primary path (because it is direct). The path over RipEX C is the first backup option (two hops) and if this path also fails, GPRS backup path is ready in case of radio failure. In cellular networks, data transfer is charged and so it is used as the last option here.

Path priorities can be changed according to our requirements. The path with the highest priority is always the primary one (the direct radio link in our example) and the path with the lowest priority is the last option (GPRS in our example).

Thanks to the Backup routes functionality, RipEX can handle various network problems without interrupting the desired network communication.

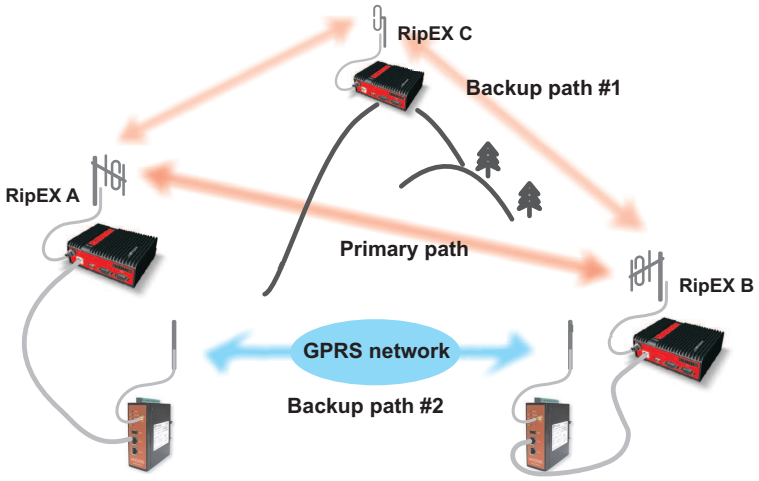


Fig. 12.1: Backup routes functionality example

Note

The Backup routes functionality can be used in the Router mode only.

The Backup routes functionality is supported by the SNMP, see Chapter 2, *SNMP* for further details.

12.2. Backup Routing Management Protocol

BRMP is the proprietary protocol developed by RACOM. It handles the Backup routing functionality in RipEX networks with respect to radio network requirements.

The protocol

- does not overload the radio network,
- enables more than one backup path,
- deals with a random packet loss and

- enables very fast path switching in cases of network failure.

The protocol always works between two particular RipEX units. Each RipEX network can contain various backup routes and each backup route consists of several alternative paths. We can even configure nested backup paths.

12.2.1. Protocol Procedure

1. RipEX A sends out “Hello” packets (UDP) via all possible paths to RipEX B.
2. RipEX B receives these packets and records them according to the received path.
3. RipEX B sends the list of received “Hello” packets within its own “Hello” packet back to RipEX A.
4. RipEX A receives this packet and evaluates the conditions of individual paths.

Individual alternative paths can obtain the following states:

Up – the path is functional and can be used.

Down – the path is not functional and cannot be used.

Unknown – the path's state cannot be evaluated due to lack of information. This state is active immediately after the RipEX power-up or its state is not being evaluated, because a higher priority path is being used.

Note

See the respective help for detailed parameter descriptions in RipEX web interface.

12.3. Configuration Examples

In this chapter, we will go through several examples in order to explain Backup routes in practice.

Please follow the examples one by one to fully understand the configuration differences and benefits of various solutions.

Note

The examples are configured similarly to the examples used in the RipEX Application note, Chapter 1, *Address planning*.

12.3.1. Radio/Radio – End Devices Connected via Serial Interface

In the first example, there are five RipEX units in a network. All end devices are connected to the RipEX units via a serial interface. It is helpful to use only the radio IP addresses for translation and data routing. Ethernet IP addresses may be assigned randomly (you can keep their defaults, however we recommend setting Ethernet addresses similar to radio IP addresses to keep things organized).

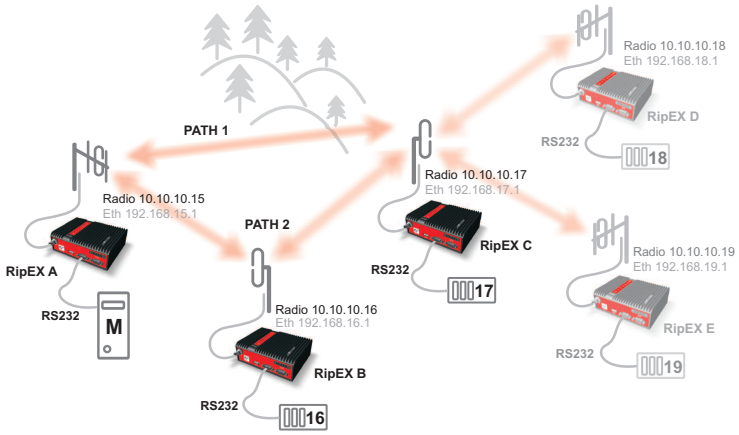


Fig. 12.2: Network topology 1

The device connected to RipEX A (10.10.10.15) is the Master station, others are slaves.

Note

We will not configure RS232 devices in this Application note

The Backup routes system can be used between RipEX A (.15) and RipEX C (.17), packets can be transmitted via:

- the primary (direct) radio link between RipEX A and RipEX C, or
- the backup (indirect) radio link over RipEX B.

See the following RipEX A routing configuration:

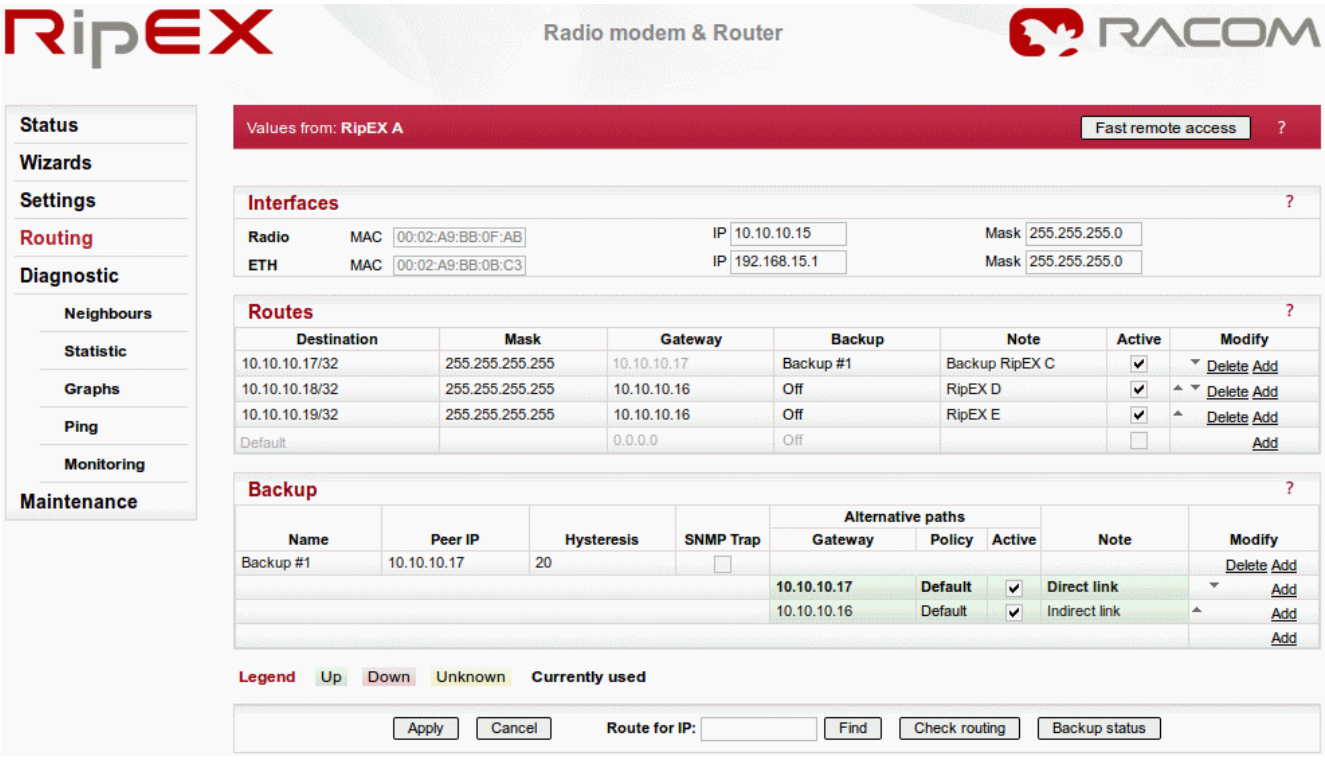


Fig. 12.3: RipEX A Routing menu – example #1

In RipEX A, we have one route which uses the backup configuration and two simple routes to other RipEX units.

The backup route is named “Backup #1” and it checks its health against the RipEX C radio IP address. The highest priority is set to the direct link and the second possibility is to use RipEX B as a repeater. Both paths are now checked by default and both are Up.

Note

Only the remote RipEX radio or the main Ethernet interface IP addresses can be used (no subnet IP addresses on RipEX Ethernet or IP of connected device behind RipEX).

See the respective configurations from RipEX B and C.

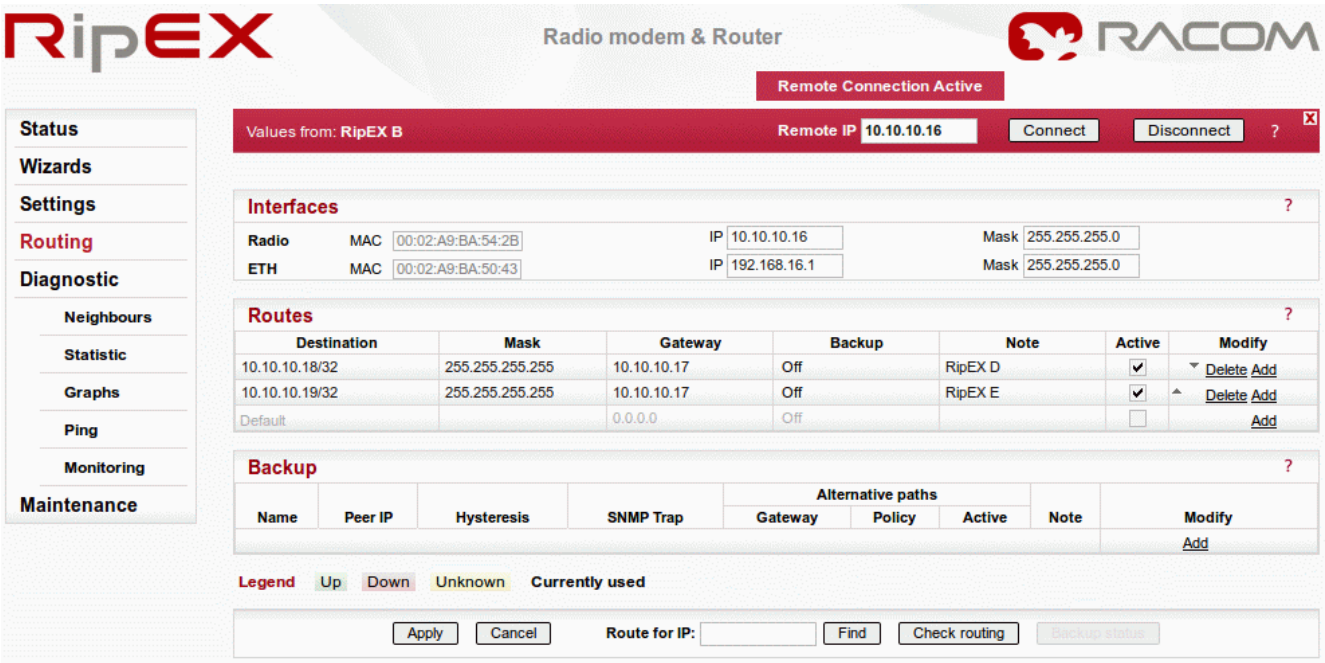


Fig. 12.4: RipEX B Routing menu – example #1

Note

RipEX B is not the end point (Peer IP) of the Backup routes system and so there is no backup route defined.

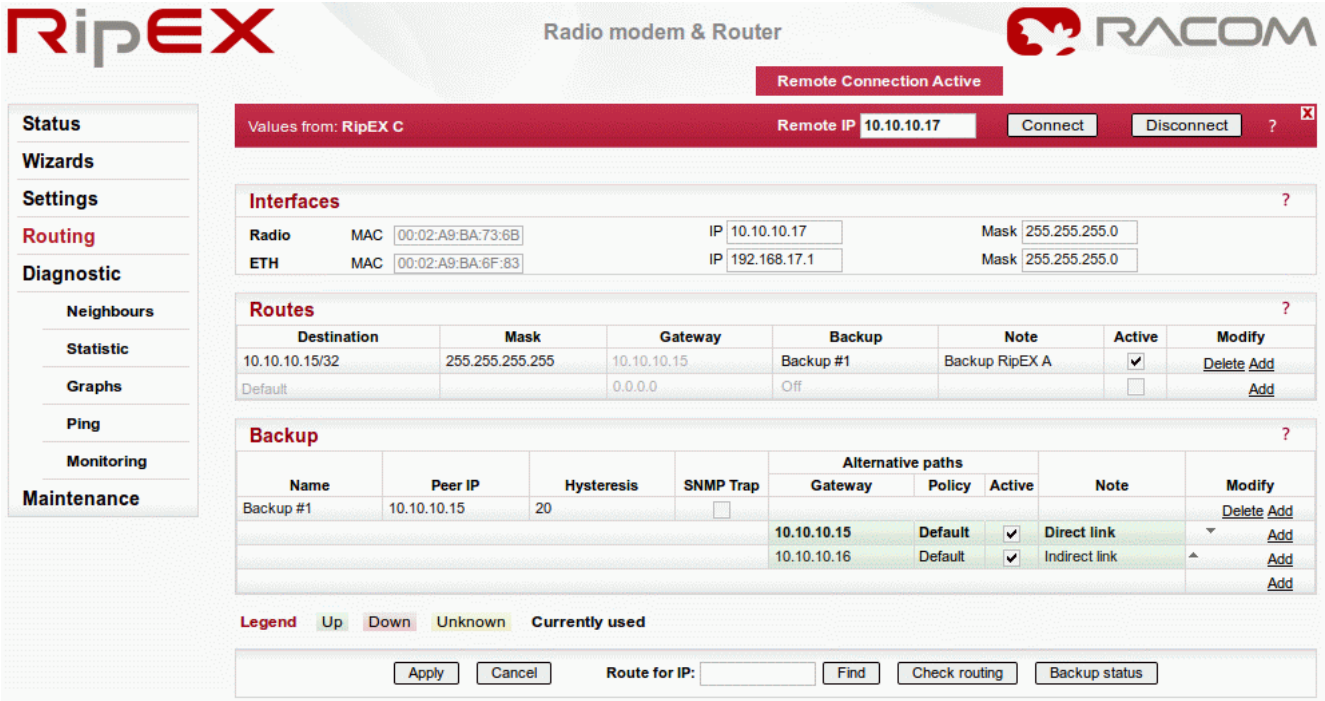


Fig. 12.5: RipEX C Routing menu – example #3

Note

See the configuration of RipEX D and E in the Application Note, Section 1.1, “End devices connected via serial interface”.

Practical Test

In this scenario, we will switch to the backup path due to a low RSS value. We must change the policy for the primary path to enable RSS checks. Click on the respective “Default” button in the **Policy** column.

Note

You can check the connectivity with a Ping feature (**Diagnostic** → **Ping**).

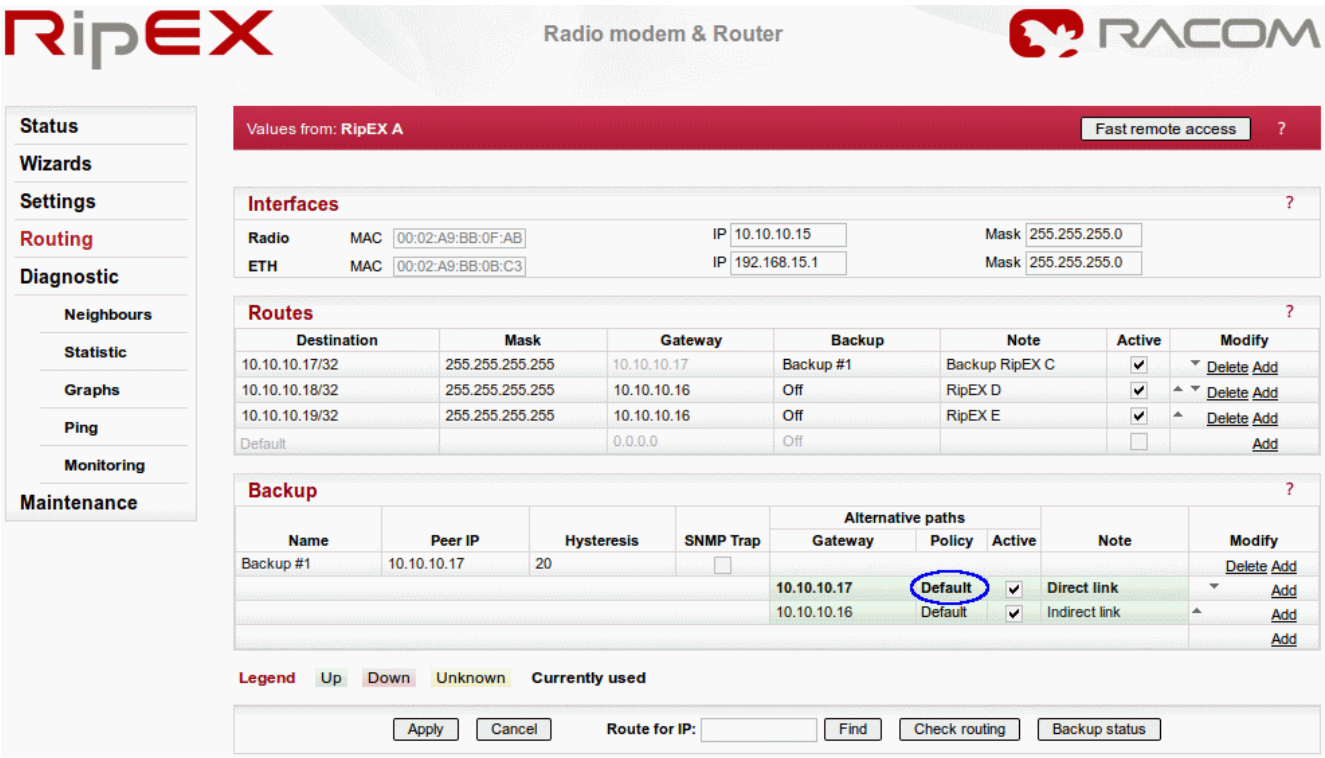


Fig. 12.6: RipEX A – Policy button

The new pop-up window appears. Change the Parameters to “Manual” and fill in the RSS [-dBm] value according to the current RSS value (see the Neighbours menu). The value needs to be higher than the current value, e.g. in the example, the current RSS value is -56 dBm. The condition for switching to the backup (indirect) path is set to -50 dBm.

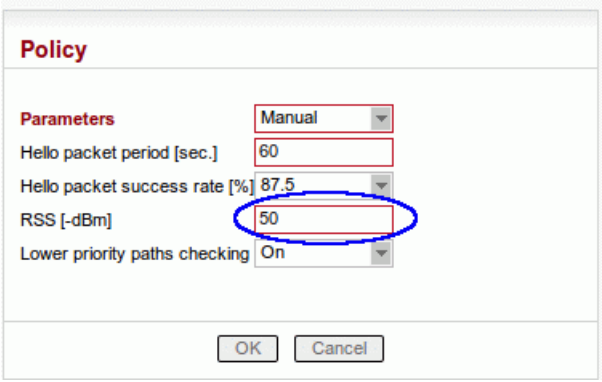


Fig. 12.7: RipEX A – Alternative path RSS change

Apply the changes and click on the Backup status button to see the changes. The policy is set to “Manual” and the backup (indirect) path is being used.

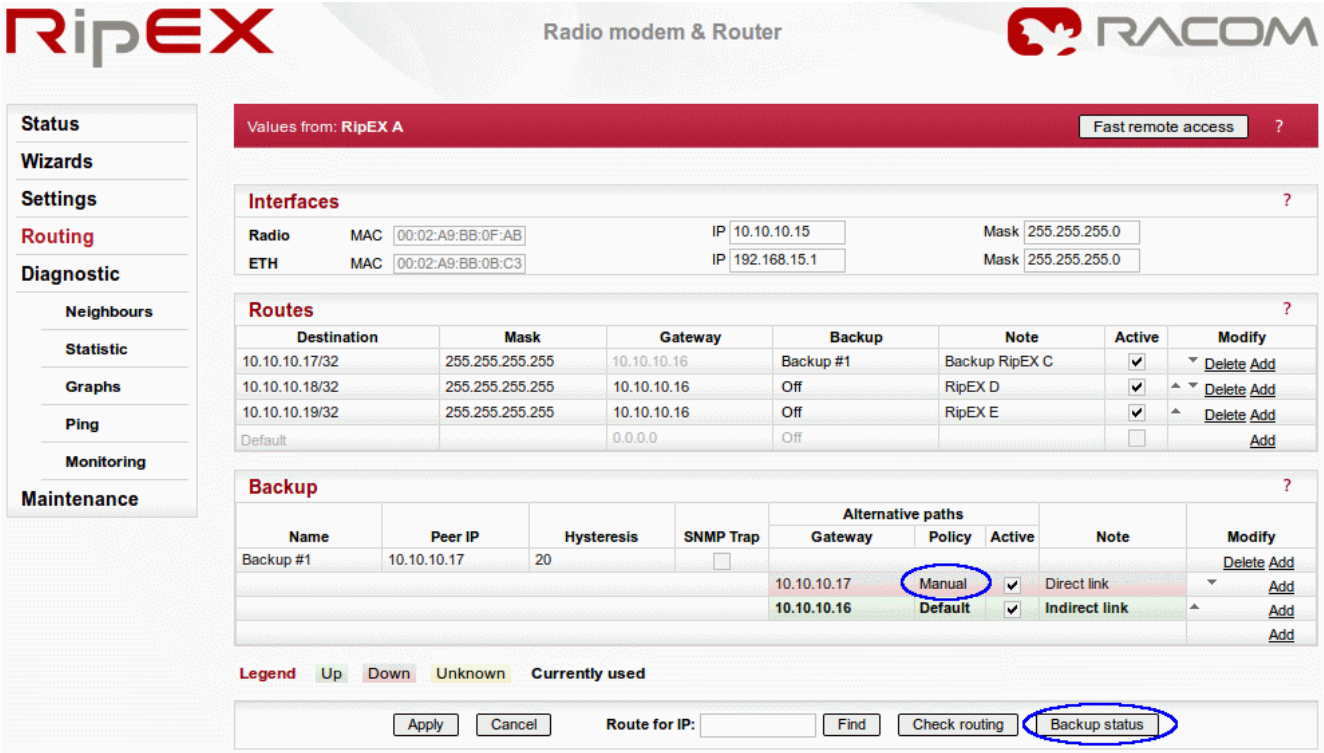


Fig. 12.8: RipEX A – Backup path is Up

Note

For proper functioning, do not forget to repeat these steps on the partner RipEX C unit. If not set on both units, RipEX A can communicate with RipEX C via the primary path in one direction and via the backup path in the other direction (asymmetric routing).

To revert to using the primary path again, disable RSS checks or improve the RSS signal between the RipEX units.

12.3.2. Radio/Radio – End Devices Connected via Ethernet Interface

In the second example, we use the same configuration except that the RTU devices are connected via the Ethernet interface. See the following diagram:

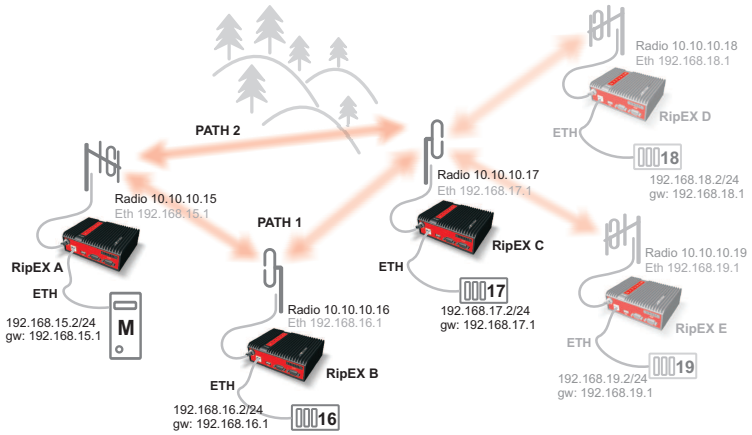


Fig. 12.9: Network topology 2

Note

In this example, we switched the priorities for the alternative paths.

RTU units are now connected via the Ethernet ports, which means we need to add the correct IP addresses and routing into the appropriate RipEX units.

If not already set, change the Ethernet IP addresses according to this topology:

- RipEX A – 192.168.15.1/24
- RipEX B – 192.168.16.1/24
- RipEX C – 192.168.17.1/24
- ...

Now we need to add the correct routing. To make the example simple, we will ignore RipEX D and RipEX E in our configuration.

See the following RipEX A routing settings:

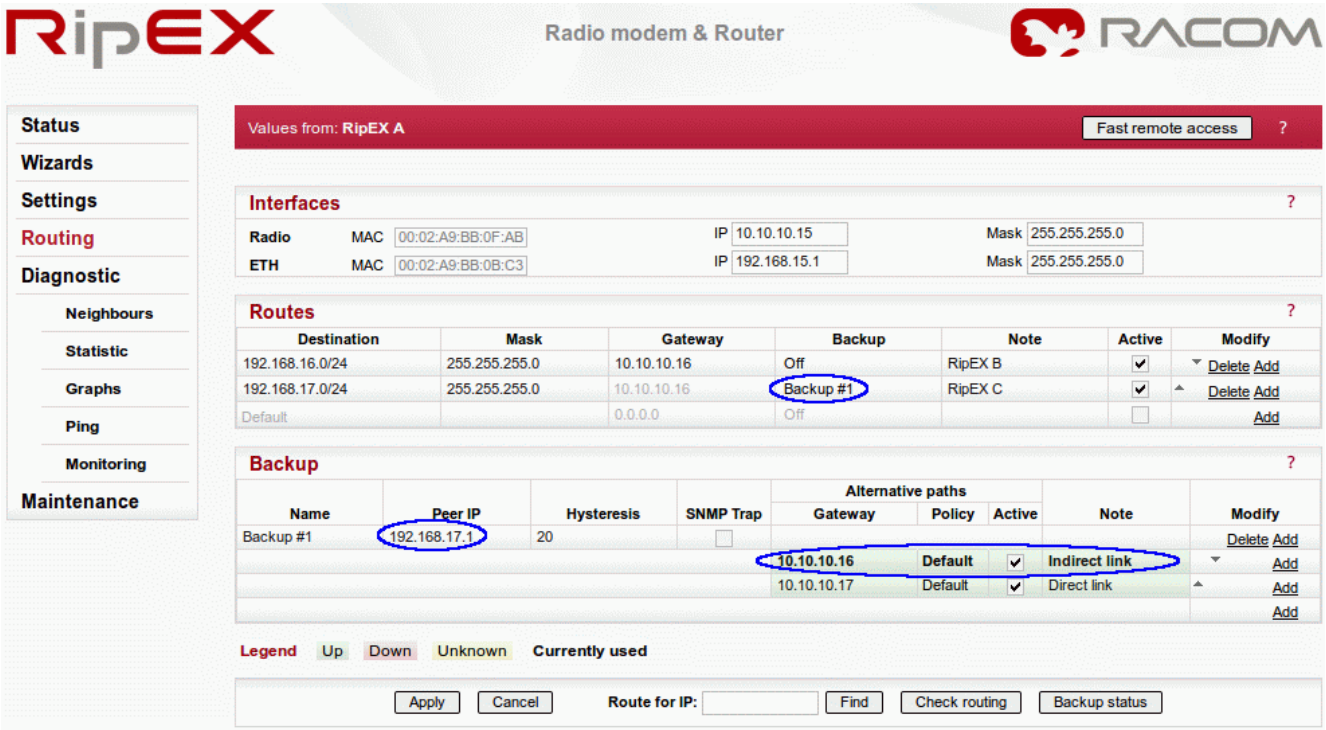


Fig. 12.10: RipEX A Routing menu – example #2

Notice that we are using the Backup routes system for the devices on the 192.168.17.0/24 network. Also notice that we filled the Peer IP with the remote RipEX Ethernet IP address. The path used currently is the primary (indirect) one, but both paths are “Up”.

Note

Only the remote RipEX radio or the main Ethernet interface IP addresses can be used (no subnet IP addresses on RipEX Ethernet or IP of connected device behind RipEX).

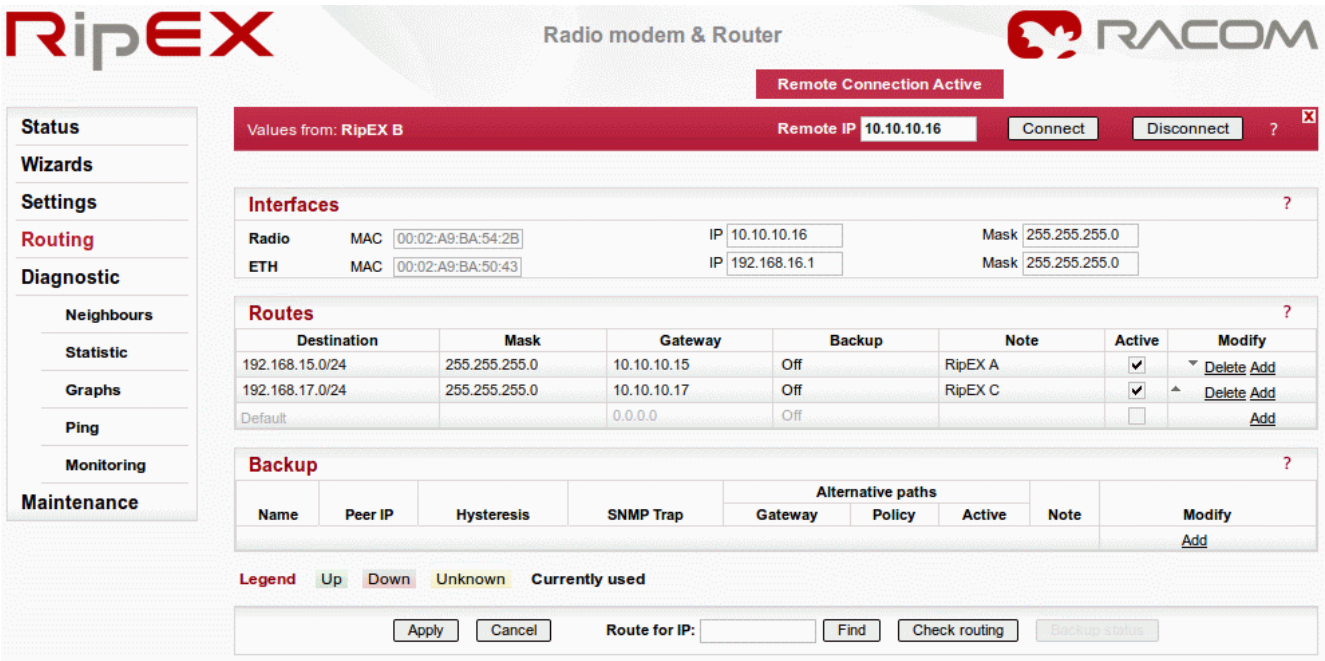


Fig. 12.11: RipEX B Routing menu – example #2

We also added paths in RipEX B for the Ethernet networks located behind other RipEX units.

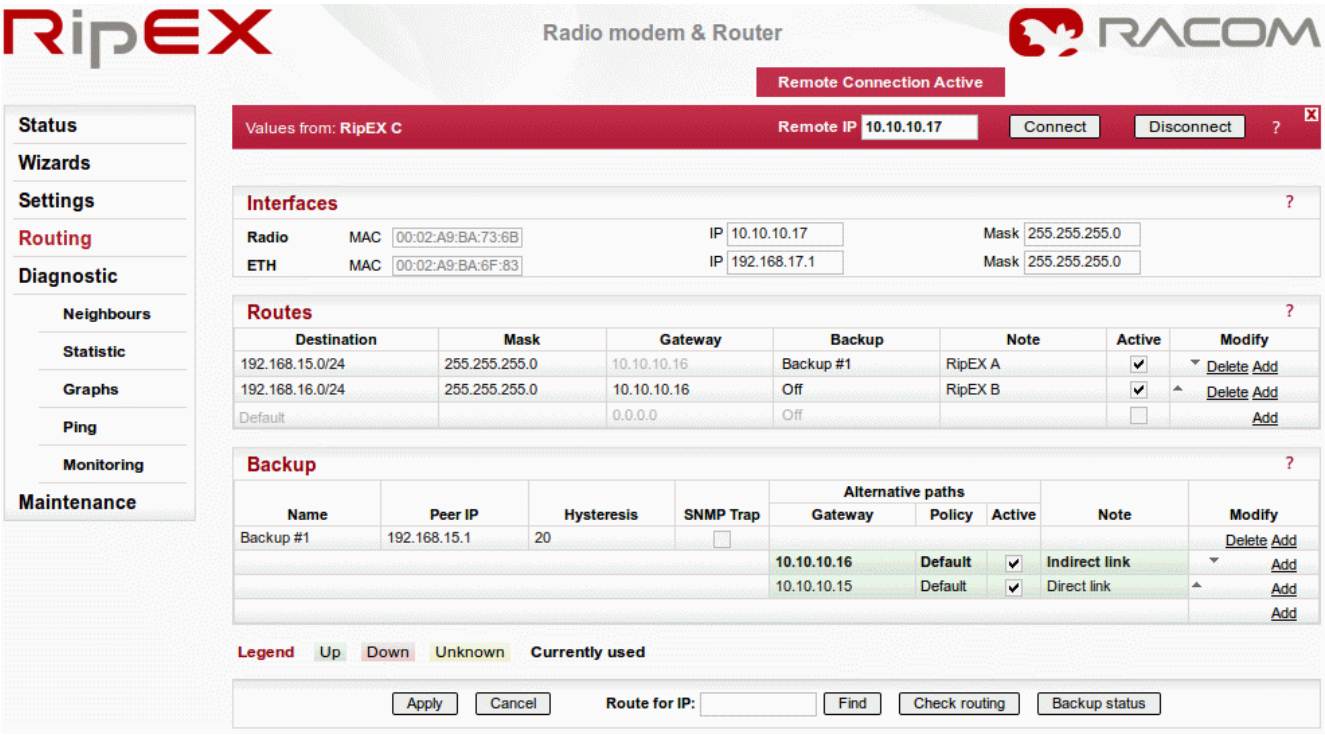


Fig. 12.12: RipEX C Routing menu – example #2

In RipEX C we have a very similar configuration to RipEX A, just in the opposite direction.

Practical Test

In this example, we will use a different method to switch between the primary and backup paths. We have set the highest priority for the indirect link (our backup path in the previous example). Whenever RipEX B is switched off, the Backup routes system will use the direct path instead.

The RipEX failure detection time is based on the Policy settings.

Note: If you set the “Hello” **packet period** to a low value (e.g. 10 seconds) and “**Hello packet success rate [%]**” to 100 %, the procedure will be very fast. But with these settings you are wasting the radio bandwidth with quite a lot of traffic and whenever a single “Hello” packet is lost, the active path is labeled as “Down”.

In the example, **we will not alter the default values**.

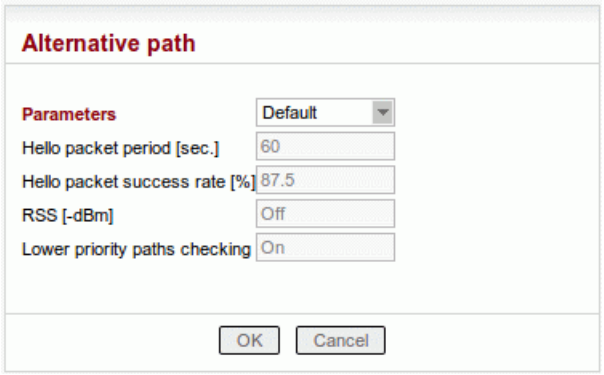


Fig. 12.13: Default Policy values

Note

“Hello packet success rate” evaluation is based on last 8 “Hello” packets.

To see the whole procedure, you can start with issuing ping packets. Go to the RipEX A **Diagnostic → Ping** menu and fill in the destination IP address (192.168.17.1). At this stage, ping packets will be successful and will be transmitted via the primary (indirect) path (e.g. check the RipEX RX/TX led diodes).

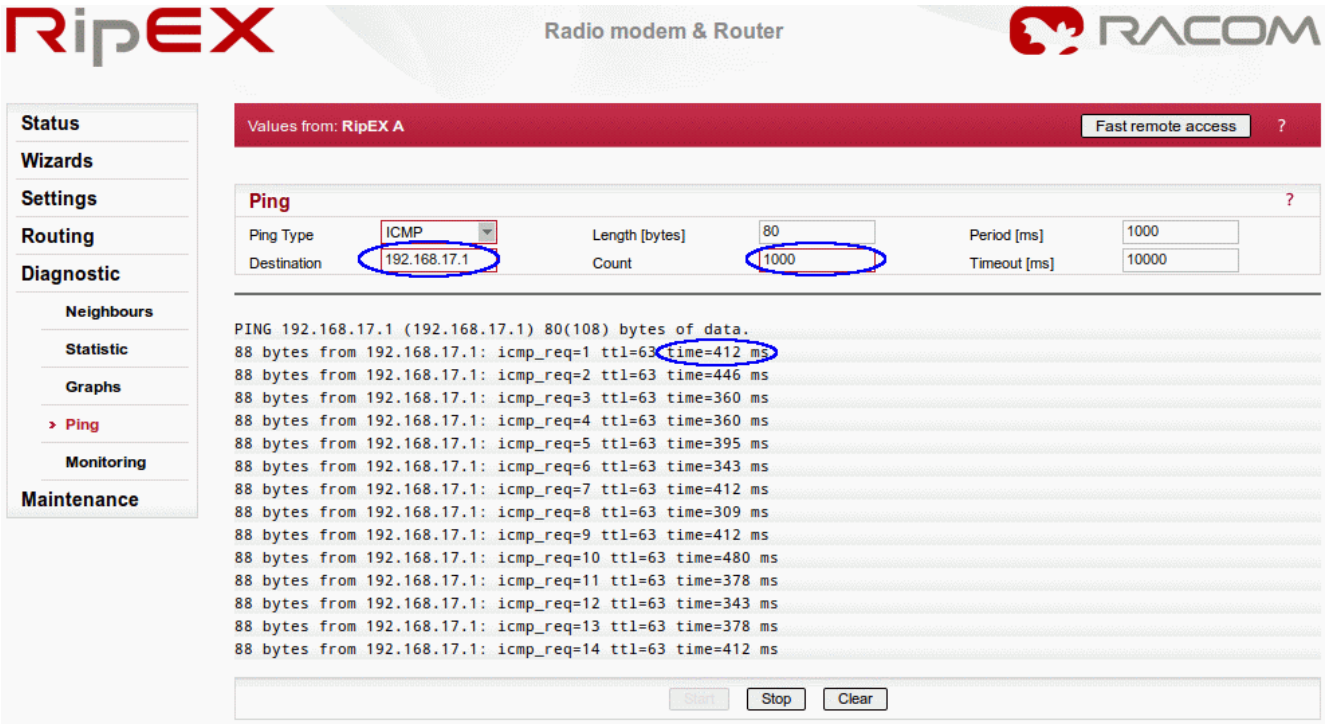


Fig. 12.14: Successful ping packets – primary (indirect) path

You can also turn on the radio interface monitoring. Go to the **Diagnostic** → **Monitoring** menu and check the radio interface. Leave other parameters at their defaults and click on the Start button. You can see all the packets in the radio network (ping packets, “Hello” packets, ARP, …).

Now turn RipEX B off, and see the differences. You can see that there are no replies to ping packets in **Ping** and **Monitoring** menu. Check the Routing menu (by pressing the Backup status button) to see when the active path is switched to the backup (direct) path.

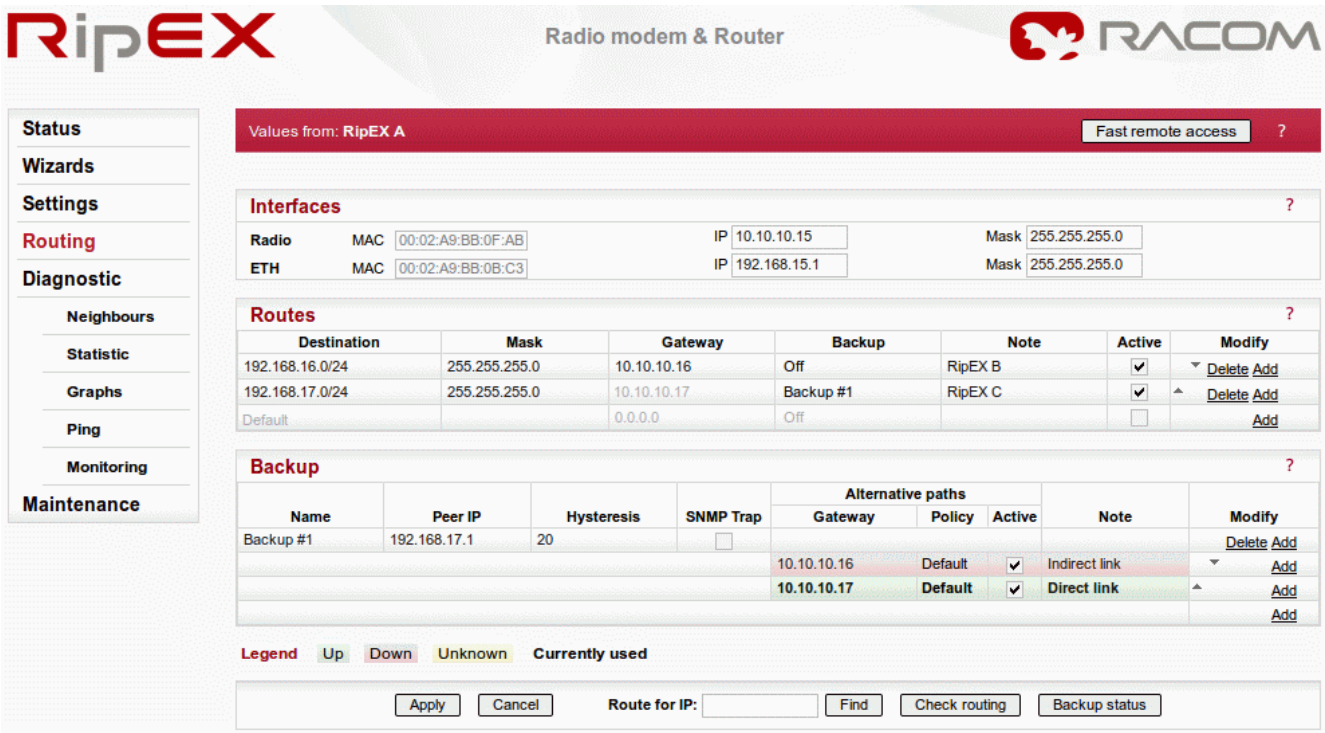


Fig. 12.15: ipEX A Routing menu – RipEX B switched off

As soon as the Backup routes system evaluates the situation correctly, the ping packets are successful again. Also notice the ping packets RTT value is lower than with the primary (indirect) path being used.

```
ping: recvmsg: No route to host
ping: recvmsg: No route to host
From 192.168.15.1: icmp_seq=558 Destination Host Unreachable
From 192.168.15.1: icmp_seq=559 Destination Host Unreachable
88 bytes from 192.168.17.1: icmp_req=563 ttl=64 time=174 ms
88 bytes from 192.168.17.1: icmp_req=564 ttl=64 time=157 ms
88 bytes from 192.168.17.1: icmp_req=565 ttl=64 time=174 ms
```

Fig. 12.16: RipEX A Ping packets – backup (direct) path

Now you can turn RipEX B back on again. Because RipEX checks the primary (indirect) path with “Hello” packets periodically, it will switch back to the primary path. This change will not cause any loss in ping packets.

12.3.3. Ethernet/Radio

In this test, the primary route is via the Ethernet link and it is backed up by the radio link.

See the following example:

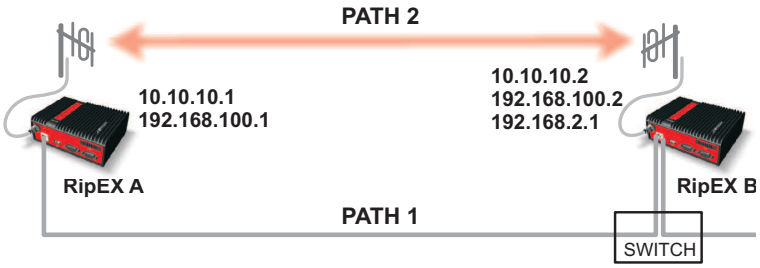


Fig. 12.17: Network topology 3

Note

This example will not be explained in as such detail as the previous ones and we will use different IP addresses.

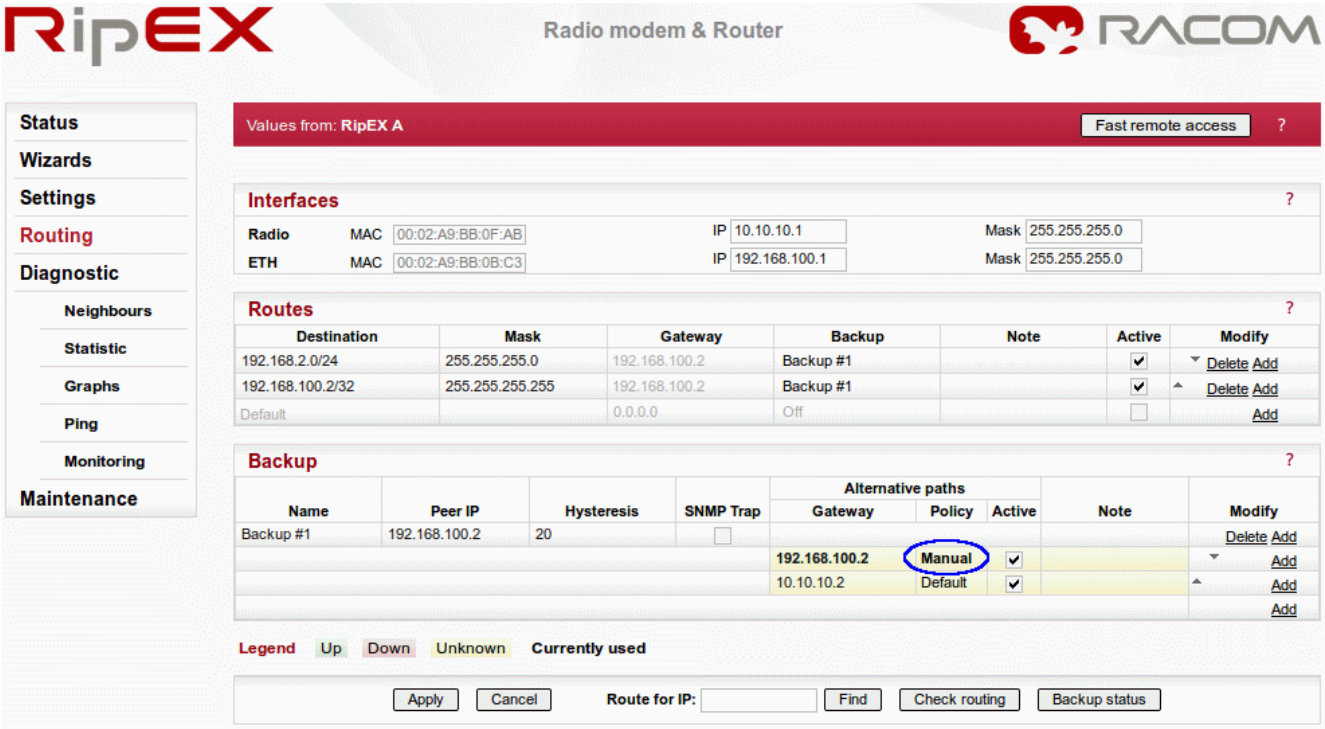


Fig. 12.18: RipEX A Routing menu – example #3

The primary Ethernet link provides a high bandwidth capacity. It is appropriate to send “Hello” packets every second. This will lead to a rapid switch over to the backup radio link in case of the Ethernet link failure.

```
ping: recvmsg: No route to host
ping: recvmsg: No route to host
From 192.168.15.1: icmp_seq=558 Destination Host Unreachable
From 192.168.15.1: icmp_seq=559 Destination Host Unreachable
88 bytes from 192.168.17.1: icmp_req=563 ttl=64 time=174 ms
88 bytes from 192.168.17.1: icmp_req=564 ttl=64 time=157 ms
88 bytes from 192.168.17.1: icmp_req=565 ttl=64 time=174 ms
```

Fig. 12.19: Hello packet period set to one second

RipEX B is configured with 192.168.100.2/24 IP address which is used only for communication between RipEX units. The additional subnet 192.168.2.0/24 is used for the rest of the Ethernet communication. See the details in ARP Proxy & VLAN Application note.

The “Hello” packet period for the Ethernet link is also set to one second on RipEX B.

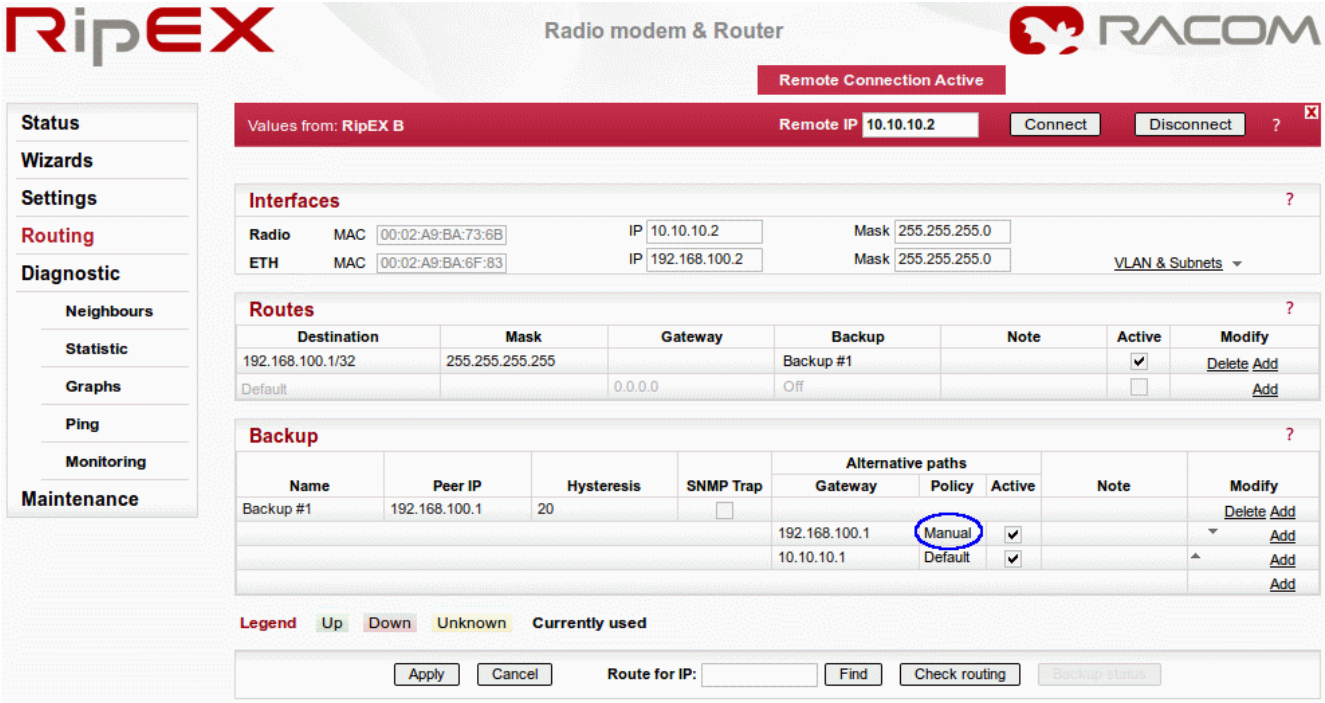


Fig. 12.20: RipEX B Routing menu – example #3

When you disconnect the primary Ethernet path, the system will automatically switch to its backup radio path. You can check this functionality using the same tools as in the previous examples.

12.4. Summary

We have described just a few basic examples of Backup routes usage. Feel free to download the RipEX User manual from http://www.racom.eu/download/hw/ripex/free/eng/ripex-m-en.pdf or the Application notes from http://www.racom.eu/download/hw/ripex/free/eng/ripex-app-en.pdf to conduct further tests.

Do not hesitate to contact us if you have any questions:

RACOM technical support team
E-mail: <support@racom.eu>
Tel.: +420 565 659 511

Appendix A. Revision History

Revision 1.1 2011-09-02
First issue

Revision 1.2 2012-01-31
New chapter – UNI protocol

Revision 1.3 2012-11-13
Modified and extended chapter – Chapter 2, *SNMP*

Revision 1.4 2013-04-20
New chapter – Chapter 11, *ARP Proxy & VLAN*

Revision 1.5 2013-04-30
New chapter – Chapter 12, *Backup routes*